

# DBD TP3

Se denomina álgebra relacional a un conjunto de operaciones simples sobre tablas, a partir de las cuales se definen operaciones más complejas mediante composición. Define, por tanto, un lenguaje de manipulación de datos.

# Como pensarlo

## Resumen de cómo pensarlo

- **Álgebra relacional:** Lo habitual es pensarlo **de afuera hacia adentro**:
  - **Primero seleccionas los registros que cumplen la condición** (los clientes con  $\text{DNI} > 22222222$ ).
  - **Después seleccionas las columnas** que quieres proyectar (nombre, apellido, etc.).

En resumen, sí puedes pensar en álgebra relacional de manera similar a `cat | cut`, primero "cortas" las filas (filtras con selección  $\sigma$ ) y luego "cortas" las columnas (con proyección  $\pi$ ).



En tu expresión relacional:

$\pi_{\text{nombre, apellido, DNI, telefono, direccion}}(\sigma_{\text{DNI} > 22222222}(\text{Cliente}))$

Primero se realiza la selección ( $\sigma$ ), que en este caso filtra las filas de la tabla `Cliente` donde el `DNI > 22222222`. Una vez hecho esto, se proyectan ( $\pi$ ) las columnas `nombre, apellido, DNI, telefono, direccion` del conjunto resultante de la selección.

En resumen, el orden de ejecución es:

1. **Selección ( $\sigma$ )**: se filtran las filas que cumplen con `DNI > 22222222`.
2. **Proyección ( $\pi$ )**: se seleccionan las columnas `nombre, apellido, DNI, telefono, direccion` de las filas filtradas.

Por lo tanto, primero se filtran las filas y luego se proyectan las columnas.



entonces en algebra relacional el orden de ejecucion es de adentro hacia afuera?



¡Exactamente! En álgebra relacional, el orden de ejecución de las operaciones es de adentro hacia afuera. Esto se debe a que las operaciones más internas generan los conjuntos de datos intermedios  $\downarrow$  a los cuales las operaciones externas se ejecutan.

# Resumen de conjuntos

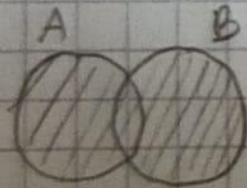
## CONJUNTOS

### UNION

$$A = \{1, 2, 3\}$$

$$B = \{3, 5, 8\}$$

$$A \cup B = \{1, 2, 3, 5, 8\}$$

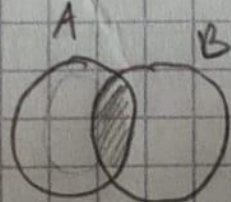


### INTERSECCION

$$A = \{1, 2, 4, 8\}$$

$$B = \{2, 5, 8\}$$

$$A \cap B = \{2, 8\}$$



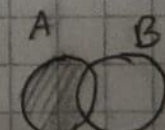
### DIFERENCIA

$$A = \{4, 5, 8\}$$

$$B = \{5, 6, 7\}$$

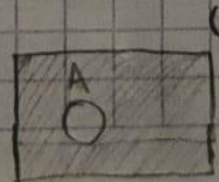
$$A - B = \{4, 8\}$$

"Los de A que no esten en B"



$$\Lambda = \emptyset$$
$$V = \Omega$$

COMPLEMENTO →

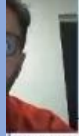


$$U = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{1, 2, 5\}$$

$$U^c = \{3, 4, 6\}$$





## Operaciones - Selección

$\sigma_p(T)$

‘ $\sigma$ ’

Produce una tabla que contiene únicamente aquellas tuplas de T que satisfacen el predicado p.

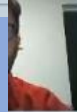
Tabla persona

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Juan	Perez
Carlos	Bilardo
Gustavo	Lopez

te da las tablas, segun la seleccion que hagas  
se pueden combinar condiciones con AND= ^ y OR=V

$\sigma_{\text{nombre}='Carlos'}(\text{persona})$

Nombre	Apellido
Carlos	Griguol
Carlos	Bilardo



## Operaciones - Proyección

$\pi_{a_1, \dots, a_n}(T)$

Produce una tabla que tiene un subconjunto de atributos de T eliminando tuplas duplicadas.

Te da las columnas seleccionadas de una tabla especificada

Tabla persona

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Juan	Perez
Carlos	Bilardo
Gustavo	Lopez

$\pi_{\text{nombre}}(\text{persona})$

Nombre
Pedro
Carlos
Juan
Gustavo

‘ $\pi$ ’



# Operaciones - Unión $T1 \cup T2$

Produce una tabla que contiene todas las tuplas de T1 más todas las de T2, eliminando tuplas duplicadas. T1 y T2 deben ser compatibles (sus esquemas deben ser equivalentes en la cantidad, posición y dominio de los atributos, aunque sus nombres sí pueden ser distintos).

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Gustavo	Lopez

Tabla tenis

Nombre	Apellido
Rafael	Nadal
Roger	Federer
Carlos	Griguol

futbol  $\cup$  tenis

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Gustavo	Lopez
Rafael	Nadal
Roger	Federer

# Operaciones - intersección

$T1 \cap T2$

Produce una tabla que contiene todas las tuplas que se encuentran tanto en T1 como en T2. T1 y T2 deben tener esquemas compatibles.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Gustavo	Lopez

Tabla tenis

Nombre	Apellido
Rafael	Nadal
Gustavo	Lopez

futbol  $\cap$  tenis

Nombre	Apellido
Gustavo	Lopez

# Ejemplo Interseccion

8. Reportar nombre, género musical y año de creación de bandas que hayan realizado recitales durante 2019, y además hayan tocado durante 2020.

$1, REZ, 1/1/2019$        $1, REZ, 1/1/2020$

2019  $\Leftarrow \pi \text{ CODIGO}, \text{NOMBRE\_BANDA}, \text{GENERO\_MUSICAL}, \text{AÑO\_CREACION} \left( \sigma \text{ FECHA} \geq 01/02/2019 \wedge \text{FECHA} \leq 31/12/2019 (\text{BANDA} \times \text{RECITAL}) \right)$

2020  $\Leftarrow \pi \text{ CODIGO}, \text{NOMBRE\_BANDA}, \text{GENERO\_MUSICAL}, \text{AÑO\_CREACION} \left( \sigma \text{ FECHA} \geq 01/02/2020 \wedge \text{FECHA} \leq 31/12/2020 (\text{BANDA} \times \text{RECITAL}) \right)$

$\pi \text{ NOMBRE\_BANDA}, \text{GENERO\_MUSICAL}, \text{AÑO\_CREACION} (2020 \cap 2019)$



¡NO TE REGALES!



# Operaciones – producto cartesiano

$$T1 \times T2$$

Produce una tabla concatenando cada tupla de T1 con todas las tuplas de T2.

Tabla futbol    Tabla tenis

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Facu	Oreja

Nombre	Apellido
Rafael	Nadal
Gustavo	Lopez

**futbol x tenis**

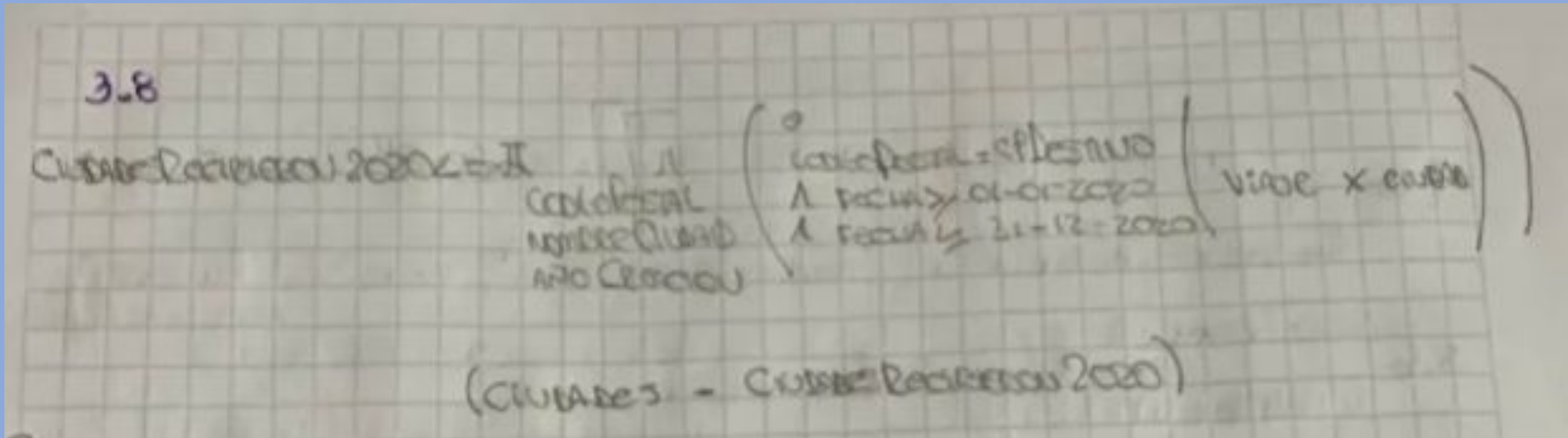
Futbol.nombre	Futbol.apellido	Tenis.nombre	Tenis.apellido
Pedro	Troglio	Rafael	Nadal
Pedro	Troglio	Gustavo	Lopez
Carlos	Griguol	Rafael	Nadal
Carlos	Griguol	Gustavo	Lopez
Facu	Oreja	Rafael	Nadal
Facu	Oreja	Gustavo	Lopez

Basicamente la table A se caga a trompadas con cada tupla de la table B

# Ejemplo product cartesiano

Listar código postal, nombre y año de creación de ciudades que no recibieron viajes durante 2020.

- Agencia (RAZON SOCIAL, dirección, telef, e-mail)
- Ciudad (CODIGOPOSTAL, nombreCiudad, añoCreación)
- Cliente (DNI, nombre, apellido, teléfono, dirección)
- Viaje (FECHA, HORA, DNI, cpOrigen (Fk), cpDestino (Fk), razon\_social(Fk), descripcion) //cpOrigen y cpDestino corresponden a la ciudades origen y destino del viaje





# Operaciones – producto natural $T1 \times T2$

Produce una tabla concatenando tuplas de ambas tablas que tengan valores iguales en atributos con igual nombre (equicombinación). Se elimina uno de los ejemplares de cada atributo común.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Facu	Oreja

Tabla tenis

Nombre	Apellido
Pedro	Troglio
Pedro	Gonzalez

**futbol  $\times$  tenis**

Futbol.Nombre	Futbol.Apellido
Pedro	Troglio

Tabla tenis

Nombre	Apellido1
Pedro	Gonzalez
Pedro	Troglio

**futbol  $\times$  tenis**

Futbol.Nombre	Futbol.Apellido	Tenis.apellido1
Pedro	Troglio	Gonzalez
Pedro	Troglio	Troglio

Por lo menos 1 nombre de una una table debe coincidir (Por lo menos 1 debe coincidir)

Si una coincide.. Debe haber un atributo que coincida en las 2 tablas... si coincide un atributo se devuelve todas las tuplas de esa fila

# Otro ejemplo mas de | X |

Cliente					
Cliente.idCliente	Cliente.nombre	Cliente.apellido	Cliente.DNI	Cliente.telefono	Cliente.direccion
1	Juan	Pérez	23456789	1234567890	Calle Falsa 123
2	Ana	Gómez	22234567	0987654321	Av. Siempreviva 742
3	Luis	Martínez	21098765	1122334455	Av. del Libertador 555
4	Sofía	López	22345678	6677889900	Calle 123

Factura				
Factura.nroTicket	Factura.total	Factura.fecha	Factura.hora	Factura.idCliente
1	50000	2020-05-20	10:00	1
1	50000	2021-05-20	11:00	4
2	70000	2020-12-15	15:00	2
3	90000	2021-01-10	11:00	1
4	45000	2020-08-22	12:00	3
5	30000	2019-06-18	09:30	4

## Cliente x Factura

Cliente.idCliente	Cliente.nombre	Cliente.apellido	Cliente.DNI	Cliente.telefono	Cliente.direccion	Factura.nroTicket	Factura.total	Factura.fecha
1	Juan	Pérez	23456789	1234567890	Calle Falsa 123	1	50000	2020-05-20
1	Juan	Pérez	23456789	1234567890	Calle Falsa 123	3	90000	2021-01-10
2	Ana	Gómez	22234567	0987654321	Av. Siempreviva 742	2	70000	2020-12-15
3	Luis	Martínez	21098765	1122334455	Av. del Libertador 555	4	45000	2020-08-22
4	Sofía	López	22345678	6677889900	Calle 123	1	50000	2021-05-20
4	Sofía	López	22345678	6677889900	Calle 123	5	30000	2019-06-18

IMPORTANTE: recalcar que antes de hacer producto natural verifiquen que solo se llamen igual los campos por los que deseen cruzar

# Ultimo ejemplo de |X|

## Reescritura sugerida:

$\pi_{nroTicket, total, fecha, hora, DNI} (\sigma_{nombreP='ProductoC'}(Producto) \bowtie Detalle \bowtie Factura \bowtie Cliente)$

### Explicación paso a paso:

1. Seleccionar el producto **Producto C**:

$\sigma_{nombreP='ProductoC'}(Producto)$

2. Unir con **Detalle**: Para obtener los tickets donde se compró el producto.

$(\sigma_{nombreP='ProductoC'}(Producto)) \bowtie Detalle$

3. Unir con **Factura**: Para obtener los detalles de las facturas asociadas a esos tickets.

$((\sigma_{nombreP='ProductoC'}(Producto)) \bowtie Detalle) \bowtie Factura$

4. Unir con **Cliente**: Para obtener la información del cliente a partir del **idCliente**.

$((\sigma_{nombreP='ProductoC'}(Producto)) \bowtie Detalle) \bowtie Factura) \bowtie Cliente$

5. **Proyección**: Finalmente, se seleccionan las columnas que necesitas: **nroTicket**, **total**, **fecha**, **hora**, **DNI**.



# Operaciones – Diferencia

T1 – T2

Produce una tabla que contiene todas las tuplas de T1 que no se encuentran en T2. T1 y T2 deben tener esquemas compatibles.

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Facu	Oreja

Tabla tenis

Nombre	Apellido
Pedro	Gonzalez
Pedro	Troglio

futbol – tenis

Nombre	Apellido
Carlos	Griguol
Facu	Oreja

Al hacer restas, laburarla para que te quede del mismo tipo  
Si restas una tablaA con datos filtrados de la misma TablaA  
asegurarte que los 2 tengan los mismos campos,  
Si meto un |x| hay mayores probabilidades de cagarla, sino  
Meto un filtro para quedarme con lo que quiero para la resta

Cliente		Factura
idCliente number		nroTicket number
nombre string		total number
apellido string		fecha string
DNI number		hora string
telefono string		idCliente number
direccion string		

**Error at line 1:** schemas are not unifiable: types are different or size is different [Cliente.idCliente : number, Cliente.nombre : string, Cliente.apellido : string, Cliente.DNI : number, Cliente.telefono : string, Cliente.direccion : string] and [Factura.nroTicket : number, Factura.total : number, Factura.fecha : string, Factura.hora : string, Factura.idCliente : number]

8. Reportar nombre y descripción de equipos que tengan inscripciones a torneos a disputarse durante 2019, pero no tienen inscripciones a torneos de 2020.

Equipo	(codigoE, nombreE, descripcionE)
Integrante	(DNI, nombre, apellido, ciudad, email, telefono, codigoE(fk))
Laguna	(nroLaguna, nombreL, ubicación, extension, descripción)
TorneoPesca	(codTorneo, fecha, hora, nroLaguna(fk), descripción)
Inscripcion	(codTorneo(fk), codigoE(fk), asistio, gano) // asistio y gano son true/false

2019  $\leftarrow \pi \text{ codigoE, nombreE, descripcionE } (\sigma \text{ FECHA} \geq 01/01/2019 \wedge \text{FECHA} \leq 31/12/2019 ((\text{Equipo} \bowtie \text{inscripcion}) \bowtie \text{TORNEOPESCA}))$

2020  $\leftarrow \pi \text{ codigoE, nombreE, descripcionE } (\sigma \text{ FECHA} \geq 01/01/2020 \wedge \text{FECHA} \leq 31/12/2020 ((\text{Equipo} \bowtie \text{inscripcion}) \bowtie \text{TORNEOPESCA}))$

$\pi \text{ nombreE, descripcionE } (2019 - 2020)$

# Operaciones – División

$T1 \div T2$

Produce una tabla con los campos de  $T1 - T2$  (están en  $T1$  y no en  $T2$ ), donde los valores en esos campos de  $T1$  se corresponden con TODAS las tuplas en  $T2$ . El esquema de  $T2$  deber estar incluido en  $T1$ .

Me retorna las columnas de los que le reste

ejem: resto futbol-tenis... a los que voy a impactar en Futbol, va ser Troglio y Gonzalez...

Ahora de esos me fijo sus tuplas y retorno eso

Tabla futbol

Nombre	Apellido
Pedro	Troglio
Carlos	Griguol
Pedro	Gonzalez

Tabla tenis

Apellido
Gonzalez
Troglio

**futbol  $\div$  tenis**

Nombre
Pedro

**PRODUCTO** = (idproducto, nombre, código barra, preciocosto)

**CLIENTE** = (idcliente, nombre, dirección, idlocalidad(FK))

**FACTURA** = (idfactura, fecha, montofactura, idcliente(FK))

**REGLON** = (idfactura, renglon, idproducto(FK), precioventa, cantidad)

**LOCALIDAD** = (idlocalidad, descripcion)

te selecciona de  $t1$ , aquellas tuplas que coinciden con TODO  $t2$

con todo hace referencia a todas las duplas de  $t2$

15:44

## ► Nombre de clientes que compraron todos los productos

Aca tengo el id, y el nombre de los clients que realizaron X compra

$$(\pi_{\text{idProducto, nombre}} (\text{Cliente} \bowtie \text{Factura} \bowtie \text{Renglon})) \div \pi_{\text{idproducto}} (\text{Producto})$$



5. Reportar nombre y descripción de equipos que tengan inscripciones en todas las lagunas.

Equipo (codigoE, nombreE, descripcionE)  
Integrante (DNi, nombre, apellido, ciudad, email, telefono, codigoE(fk))  
Laguna (nroLaguna, nombreL, ubicación, extension, descripción)  
TorneoPesca (codTorneo, fecha, hora, nroLaguna(fk), descripcion)  
Inscripcion (codTorneo(fk).codigoE(fk), asistio, gano) // asistio y gano son true/false

LAGUNAS  $\leftarrow \pi$  NROLAGUNA(LAGUNA)

ARDOL  $\leftarrow \pi$  NOMBREE, DESCRIPCIONE, NROLAGUNA(EQUIPO | X | INSCRIPCION | X | TORNEOPESCA | X | LAGUNA)

ARDOL o/o LAGUNAS

Futbol	
Futbol.Nombre	Futbol.Apellido
Pedro	Troglio
Carlos	Griguol
Eduardo	Gonzalez
Eduardo	Troglio
Pedro	Gonzalez

Tenis1

Tenis1.Apellido

Gonzalez

Troglio

Futbol ÷ Tenis1

Futbol.Nombre

Pedro

Eduardo

# Operaciones – Renombre

$\rho T$  (Table)

Renombra la tabla Table a T.

# Operaciones – Asignación

$A \leftarrow \text{Consulta}$

Vuelca a A los resultados de CONSULTA. Luego puedo utilizar A.

# Ejemplo clave, de renombre

Agencia (RAZON SOCIAL, dirección, telef, e-mail)

Ciudad (CODIGOPOSTAL, nombreCiudad, añoCreación)

Cliente (DNI, nombre, apellido, teléfono, dirección)

Viaje (FECHA, HORA, DNI, cpOrigen (Fk), cpDestino (Fk), razon\_social(Fk), descripcion) //cpOrigen y cpDestino corresponden a la ciudades origen y destino del viaje

6. Listar datos personales de clientes que viajaron con destino a la ciudad de 'Lincoln' pero no realizaron viajes con origen en 'La Plata'.

6. Listar datos personales de clientes que viajaron con ciudad destino 'Lincoln' pero no realizaron viajes con origen 'La Plata'.

$LaPlata \leftarrow \pi_{CODIGOPOSTAL} (\sigma_{nombreCiudad = "La Plata"} (Ciudad))$

$Lincoln \leftarrow \pi_{CODIGOPOSTAL} (\sigma_{nombreCiudad = "Lincoln"} (Ciudad))$

$ViajesLincoln \leftarrow \sigma_{cp\overset{destino}{\cancel{Origen}} = Lincoln} (Viaje \mid x \mid Cliente)$

$ViajesLaPlata \leftarrow \sigma_{cpOrigen = LaPlata} (Viaje \mid x \mid Cliente)$

$\pi_{DNI, nombre, apellido, telefono, direccion} (ViajesLincoln - ViajesLaPlata)$



# GUARDA CON LA INTEGRIDAD REFERENCIAL

## Actualización de tablas

Producto=(codProd, desc, existAct, existMin, pVAct)

- ❑ Incorporar el producto (1235, “tuerca de 9 mm”, 10, 50,\$10):  
$$\text{Producto} \leftarrow \text{Producto} \cup \{(1235, \text{“tuerca de 9 mm”}, 10, 50, \$10)\}$$

Al hacer eliminaciones de datos, TENER CUIDADO CON LA INTEGRIDAD REFERENCIAL

- ❑ Eliminar el producto 893:

$$\text{Producto} \leftarrow \text{Producto} - \sigma_{\text{codProd}=893}(\text{Producto})$$

- ❑ Aumentar el 1% el precio de venta actual de todos los productos:

$$\delta \text{ pVAct} \leftarrow \text{pVAct} * 1,01(\text{Producto})$$

Actualizame => simboliza esto

Agencia (RAZON SOCIAL, dirección, telef, e-mail)  
Ciudad (CODIGOPOSTAL, nombreCiudad, añoCreación)  
Cliente (DNI, nombre, apellido, teléfono, dirección)  
Viaje (FECHA, HORA, DNI, cpOrigen (Fk), cpDestino (Fk), razon\_social(Fk), descripcion) //cpOrigen y cpDestino corresponden a la ciudades origen y destino del viaje

## 1. Eliminar el cliente con DNI:25326992.



Sí, la **integridad referencial** es importante al eliminar un cliente en una base de datos relacional, y en álgebra relacional también debemos considerar este aspecto si se está haciendo la operación en un contexto con claves foráneas.

En este caso:

- La tabla **Viaje** tiene una clave foránea que referencia a la tabla **Cliente** mediante el atributo **DNI**.
- Al eliminar un cliente, si ese **DNI** está referenciado en la tabla **Viaje**, primero deberíamos gestionar los registros asociados en dicha tabla, ya que eliminar un cliente sin manejar las referencias rompería la integridad referencial.

### Manejo de integridad referencial en álgebra relacional:

Tienes dos opciones principales:

1. **Eliminación en cascada:** Elimina todos los viajes relacionados con el cliente antes de eliminar el cliente.

En álgebra relacional:

1. Eliminar primero los viajes relacionados con el cliente:

$$Viaje_{nuevo} = Viaje - \sigma_{DNI=25326992}(Viaje)$$

2. Luego, eliminar al cliente:

$$Cliente_{nuevo} = Cliente - \sigma_{DNI=25326992}(Cliente)$$



# CRUD

La operación de alta permite agregar una nueva tupla a una tabla existente. Para ello, se debe realizar una operación de unión entre la tabla y una tupla escrita de manera literal.

Alumnos  $\leftarrow$  Alumnos U {"Delia", "30777987", 2, 1}

## Modificaciones

A diferencia de las dos operaciones de actualización anteriores, que se resolvían a partir de la utilización de operadores básicos, la operación de modificación necesita representar el cambio con un operador,  $\delta$ . El formato genérico de la operación es:

$$\delta_{\text{atributo} = \text{valor\_nuevo}}(\text{tabla})$$

donde  $\delta$  es el operador que indica modificación, tabla es la tabla que se modifica, atributo es el atributo de la tabla a modificar y valor\_nuevo se asigna como nuevo contenido del atributo.

Ejemplo 15: Se debe modificar el número de DNI de Pettorutti por 34567231.

$$\delta_{\text{dni} = "34567231"}(\sigma_{\text{nombre} = "Pettorutti"}(\text{alumnos}))$$

Nótese que, previo a realizar la modificación, se debe seleccionar la tupla sobre la que se desea hacer el cambio. Si no se hiciera de esa forma, todos los DNI de todos los alumnos de la tabla se modificarían.

La operación de baja permite quitar una tupla de una tabla. Para ello, se debe realizar una operación de diferencia entre la tabla y una tupla escrita de manera literal.

Ejemplo 14: Se debe quitar de la tabla alumnos al alumno López.

Alumnos  $\leftarrow$  Alumnos  $- \sigma_{\text{nombre} = "Lopez"}(\text{alumnos})$



# CONSULTAS LOS PUENTES, que tipo de puentes son validos, para saltar entre fk a pk

**PRODUCTO** = (idproducto, nombre, código barra, preciocosto)

**CLIENTE** = (idcliente, nombre, dirección, idlocalidad(FK))

**FACTURA** = (idfactura, fecha, montofactura, idcliente(FK))

**REGLON** = (idfactura, renglon, idproducto(FK), precioventa, cantidad)

**LOCALIDAD** = (idlocalidad, descripcion)

► Dar de baja aquellos productos que nunca fueron facturados.

$\text{Producto} \leftarrow \pi_{\text{idproducto, nombre, codigobarra, preciocosto}} (\text{Factura} \bowtie \text{Renglon} \bowtie \text{Producto})$

# Ejemplo práctico

**PRODUCTO** = (idproducto, nombre, código barra, preciocosto)

**CLIENTE** = (idcliente, nombre, dirección, idlocalidad(FK))

**FACTURA** = (idfactura, fecha, montofactura, idcliente(FK))

**REGLON** = (idfactura, renglon, idproducto(FK), precioventa, cantidad)

**LOCALIDAD** = (idlocalidad, descripcion)

- Dar de baja las facturas del cliente 'Ramon Perez'.

$$\text{FacturasDPerez} \leftarrow \pi_{\text{idfactura, fecha, montofactura, idCliente}} \text{Factura} \mid \sigma_{(\text{nombre} = \text{'Ramon Perez'})} \text{Cliente}$$

$$\text{RenglonesDPerez} \leftarrow \pi_{\text{idfactura, renglon, idProducto, precioVenta, cantidad}} \text{FacturasDPerez} \mid \text{Renglon}$$

$$\text{Renglones} \leftarrow \text{Renglones} - \text{RenglonesDPerez}$$

# Pregunta

Banda (codigoB, nombreBanda, genero\_musical, año\_creacion)

Integrante (DNI, nombre, apellido, dirección, email, fecha\_nacimiento, codigoB(fk))

Escenario (nroEscenario, nombre\_escenario, ubicación, cubierto, m2, descripción)

Recital (fecha, hora, nroEscenario(fk), codigoB(fk) )

$R_{RATONES} \leftarrow \sigma_{R.NOMBREBANDA = 'RATONES PARANOICOS' (BANDA)}$

$\int R_{RATONES}.AÑO\_CREACION \leftarrow R_{RATONES}.AÑO\_CREACION = 1983$

2.7: Modificar el año de creación de la banda de nombre 'Ratones Paranoicos' a: 1983

$\delta_{año\_creacion = 1983} (\sigma_{nombreBanda = "Ratones Paranoicos" (banda)})$

$\int AÑO\_CREACION = 1983 (\sigma_{B.NOMB-BANDA = 'RATONES PARANOICOS' (BANDA)})$

$COMPLE \leftarrow [ \sigma_{NOMBREBANDA = "RATONES PARANOICOS" (BANDA)} ]$

$\delta_{BANDA.AÑO\_CREACION \leftarrow COMPLE.AÑO\_CREACION = 1983}$



- Equipo (codigoE, nombreE, descripcionE)  
Integrante (DNI, nombre, apellido, ciudad, email, telefono, codigoE(fk))
- Laguna (nroLaguna, nombreL, ubicación, extension, descripción)
- TorneoPesca (codTorneo, fecha, hora, nroLaguna(fk), descripcion)
- Inscripcion (codTorneo(fk), codigoE(fk), asistio, gano) // asistio y gano son true/false

4. Listar para la laguna con nombre 'laguna x', nombre y descripción de equipos ganadores de torneos que se disputaron durante 2019 en la mencionada laguna.

ganadores  $\leftarrow \sigma_{\text{gano}=\text{true}}(\text{Inscripcion})$

lagunaX  $\leftarrow \sigma_{\text{nombreL}=\text{"laguna x"}}(\text{Laguna})$

torneos  $\leftarrow \sigma_{\text{fecha} \geq 01/01/2019 \text{ and } \text{fecha} \leq 31/12/2019}(\text{TorneoPesca})$

$\pi_{\text{nombreE}, \text{descripcionE}}(\text{Equipo} \mid x \mid \text{ganadores} \mid x \mid \text{torneos} \mid x \mid \text{lagunaX})$

## Enunciado:

"Listar el nombre y precio de aquellas partes que figuren en todos los service realizados durante 2019."

## Desglose:

### 1. "Partes":

- Se refiere a las piezas o componentes (Parte) que pueden ser utilizadas durante el mantenimiento o reparación de vehículos.

### 2. "Service":

- El service es el mantenimiento o revisión que se le hace a un vehículo en una fecha específica. Esto está representado por la tabla Service, donde se guardan los datos de cada service realizado, como la fecha, la patente del vehículo, etc.

### 3. "Todos los service realizados durante 2019":

- Aquí el enunciado se refiere a todos los services que ocurrieron entre el 1 de enero de 2019 y el 31 de diciembre de 2019. Es decir, en el año 2019, puede haber habido varios services, y necesitas considerar cada uno de esos services.

### 4. "Partes que figuren en todos los service":

- Ahora, lo más importante es que te están pidiendo encontrar las partes (componentes o piezas) que hayan sido utilizadas en todos los services de 2019, sin excepciones. Es decir, si un service usó una parte, pero otro service no la usó, esa parte no debe aparecer en la lista final.
- Solo las partes que se hayan utilizado en cada uno de los services que ocurrieron en 2019 serán parte de la respuesta.

### 5. "Nombre y precio":

- Finalmente, una vez que hayas identificado las partes que estuvieron en todos los services de 2019, necesitas mostrar el nombre y el precio de esas partes.

## Ejemplo simplificado:

Imagina que en 2019 se realizaron 3 services:

- Service 1 en enero.
- Service 2 en mayo.
- Service 3 en septiembre.

Cada service puede haber utilizado diferentes partes. Por ejemplo:

- El Service 1 utilizó las partes A, B y C.
- El Service 2 utilizó las partes A y C.
- El Service 3 utilizó las partes A, B y C.

En este caso, la única parte que aparece en todos los services es la parte A. Las partes B y C no se usaron en todos los services, así que no se incluirían en el resultado final.

## Objetivo final:

- Identificar aquellas partes que se usaron en todos los services de 2019.
- Listar el nombre y precio de esas partes.

## Concepto clave:

Estás buscando partes que sean comunes a todos los services del año 2019, y una vez que las tengas, mostrar su nombre y precio.



4)  $FECHA \geq 01/01/2019$   
 $SERVICE2019 \leftarrow \sigma_{FECHA \leq 31/12/2019} (SERVICE)$

$CODPARTES2019 \leftarrow \pi_{CODPARTE} (SERVICE2019 \bowtie SERVICEPARTE)$

$SERVICIOS2019CONCODP \leftarrow \pi_{CODPARTE, NOMBRE, PRECIOPARTE} ((SERVICE2019 \bowtie SERVICEPARTE) \bowtie PARTE)$

$SERVICIOS2019CONCODP \text{ o/o } CODPARTES$

Vehiculo = (patente, modelo, marca, peso, km)

Camion = (patente(fk), largo, max\_toneladas, cant\_ruedas, tiene\_acoplado)

Auto = (patente(fk), es\_electrico, tipo\_motor)

Service = (fecha, patente(fk), km\_service, observaciones, monto)

Parte = (cod\_parte, nombre, precio\_parte)

Service\_Parte = ([fecha, patente](fk), cod\_parte(fk), precio, cantidad)

1. Listar todos los datos de aquellos camiones que tengan entre 8 y 12 ruedas, y que hayan realizado algún service antes de los 10000 km.
2. Listar los autos que hayan realizado el service "cambio de aceite" antes de los 13.000 km o hayan realizado el service "inspección general" que incluya la parte "filtro de combustible".
3. Dar de baja todos los camiones con más de 350.000 km.
4. Listar el nombre y precio de aquellas partes que figuren en todos los service realizados durante 2019.



Vehiculo = (patente, modelo, marca, peso, km)

Camion = (patente(fk), largo, max\_toneladas, cant\_ruedas, tiene\_acoplado)

Auto = (patente(fk), es\_electrico, tipo\_motor)

Service = (fecha, patente(fk), km\_service, observaciones, monto)

Parte = (cod\_parte, nombre, precio\_parte)

Service\_Parte = ([fecha, patente](fk), cod\_parte(fk), precio, cantidad)

Elimina toda la informacion, relacionada a camiones....DELETE ALL

3. Dar de baja todos los camiones con más de 350.000 km.

ELIMINA TODOS SERVICIOS RELACIONADOS A UNA PATENTE... PERO NO LOS DATOS DEL AUTO

7. Dar de baja todos los services que se realizaron al auto con patente 'AAA564'.

**4. Listar nombre, descripción, precio y stock de productos **no vendidos** a clientes que tengan teléfono con característica: 221 (La característica está al comienzo del teléfono). Ordenar por nombre.**

```
SELECT p.nombre, p.descripcion, p.precio, p.stock
```

```
FROM Producto p
```

```
EXCEPT (equivalente a la resta en álgebra)
```

```
(SELECT p.nombre, p.descripcion, p.precio, p.stock
```

```
FROM ((Cliente c NATURAL JOIN factura f) NATURAL JOIN detalle d) NATURAL JOIN producto p
```

```
WHERE (c.telefono LIKE '221%')
```

```
)
```

**10. Listar DNI, apellido y nombre de clientes donde el monto total comprado, teniendo en cuenta todas sus facturas, supere \$10.000.000.**

```
SELECT c.nombre, c.apellido, c.DNI, c.teléfono, c.dirección
```

```
FROM (Factura f NATURAL JOIN Cliente c)
```

```
GROUP BY c.idCliente, c.dni, c.nombre
```

```
HAVING SUM(f.total) > 10.000.000
```

**6. Listar nombre, apellido, DNI, teléfono y dirección de clientes que compraron los productos con nombre 'prod1' y 'prod2' pero nunca compraron el producto con nombre 'prod3'.**

```
SELECT c.nombre, c.apellido, c.DNI, c.teléfono, c.dirección
```

```
FROM ((Cliente c NATURAL JOIN factura f) NATURAL JOIN detalle d) NATURAL JOIN producto p
```

```
WHERE (p.nombreP = 'prod1') and (c.idCliente IN (
```

```
    SELECT c.idCliente
```

```
    FROM ((Cliente c NATURAL JOIN factura f) NATURAL JOIN detalle d) NATURAL JOIN producto p
```

```
    WHERE (p.nombreP = 'prod2'))
```

```
EXCEPT
```

```
(SELECT c.nombre, c.apellido, c.DNI, c.teléfono, c.dirección
```

```
FROM ((Cliente c NATURAL JOIN factura f) NATURAL JOIN detalle d) NATURAL JOIN producto p
```

```
WHERE (p.nombreP = 'prod3'))
```

Vehiculo = (patente, modelo, marca, peso, km)  
 Camion = (patente(fk), largo, max\_toneladas, cant\_ruedas, tiene\_acoplado)  
 Auto = (patente(fk), es\_electrico, tipo\_motor)  
 Service = (fecha, patente(fk), km\_service, observaciones, monto)  
 Parte = (cod\_parte, nombre, precio\_parte)  
 Service\_Parte = ([fecha, patente](fk), cod\_parte(fk), precio, cantidad)

3)

TARGET  $\leftarrow \pi_{PATENTE} (\sigma_{KM > 350.000} (CAMION \mid X \mid VEHICULO))$   
 VEHICULO  $\leftarrow VEHICULO - (TARGET \mid X \mid VEHICULO)$   
 SERVICE  $\leftarrow SERVICE - (SERVICE \mid X \mid TARGET)$   
 SERVICE\_PARTE  $\leftarrow SERVICE\_PARTE - (SERVICE\_PARTE \mid X \mid TARGET)$   
 CAMION  $\leftarrow CAMION - (CAMION \mid X \mid TARGET)$

3. Dar de baja todos los camiones con más de 350.000 km.

Camiones350000  $\leftarrow \sigma_{(km > 350000)} (Camión \mid x \mid Vehiculo)$

Service350000  $\leftarrow (\pi_{fecha, patente, km\_service, observaciones, monto} (\rho_s Service \mid x \mid \rho_c Camiones350000))$

ServiceParte350000  $\leftarrow (\pi_{fecha, patente, cod\_parte, precio} (\rho_{sp} ServiceParte \mid x \mid \rho_c Camiones350000))$

- Dar de alta una parte, cuyo nombre sea "Aleron" y precio \$3400.
- Dar de baja todos los services que se realizaron al auto con patente 'AAA564'.
- Modificar el precio de las partes incrementando un 15% dicho valor.
- Listar todos los vehículos que hayan tenido services durante el 2019.

1.2)

PARTE  $\leftarrow PARTE \cup \{(1234, 'ALERON', 3400)\}$   
 SUPONER QUE EL COD 1234 ES UNICO  
 7) SERVICE\_PARTE  $\leftarrow SERVICE\_PARTE - (\sigma_{PATENTE = 'AAA564'} (SERVICE\_PARTE))$   
 SERVICE  $\leftarrow SERVICE - (\sigma_{PATENTE = 'AAA564'} (SERVICE))$   
 8)  $\sigma_{PRECIO\_PARTE} \leftarrow PRECIO\_PARTE * 1,15 (PARTE)$   
 9) SERVICES\_2019  $\leftarrow \sigma_{FECHA \geq 01/01/2019 \wedge FECHA \leq 31/12/2019} (SERVICE)$   
 $\pi_{V.PATENTE, MODELO, MARCA, PESO, KM} (SERVICES_{2019} \mid X \mid VEHICULO)$