# Ayudantía 4

IIC2413 — Bases de Datos

## Notación (recordatorio)

- Para una relación $R$, $\mathsf{att}(R)$ denota su conjunto de atributos.
- **Operadores base del álgebra relacional** (para relaciones $R, S$, atributos $A, B$ y constante $a \in \mathbf{Data}$):
  - $R \cup S,\ R \cap S,\ R \setminus S$ (definidos si $\mathsf{att}(R) = \mathsf{att}(S)$)
  - $\sigma_{A=a}(R),\ \sigma_{A=B}(R)$, y su generalización $\sigma_\theta(R)$ con $\theta$ fórmula booleana de igualdades $(\wedge, \vee, \neg)$, además de $\sigma_{A\sim a}, \sigma_{A\sim B}$ con $\sim \in \{<, >, \leq, \geq\}$.
  - $\rho_{A\to B}(R)$ (definido si $A \in \mathsf{att}(R)$, $B \notin \mathsf{att}(R)$)
  - $\pi_{B_1,\ldots,B_k}(R)$
  - $R \bowtie S$ (join natural), $R \times S$ (caso particular de $\bowtie$ cuando $\mathsf{att}(R) \cap \mathsf{att}(S) = \varnothing$).
- **Consultas de primer orden**: $Q(x_1, \ldots, x_n) \leftarrow \varphi$, donde $\varphi$ es una fórmula con $\mathsf{FV}(\varphi) \subseteq \{x_1, \ldots, x_n\}$ construida a partir de $x = a$, $x = y$, $R(t_1, \ldots, t_k)$ mediante $\wedge, \vee, \neg, \exists x, \forall x$. Recuerde que $\forall x.\, \varphi \equiv \neg\exists x.\, \neg\varphi$, clave para traducir consultas del tipo *"para todos"/"ninguno"*.

### Recordatorio: SQL, agregación y nulos

- Plantilla general: `SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ...`. Las funciones de agregación (`COUNT`, `SUM`, `AVG`, `MIN`, `MAX`) **no existen** en álgebra relacional ni en consultas de primer orden: cualquier consulta que las use excede el poder expresivo de ambos formalismos.
- `EXISTS`/`NOT EXISTS` (correlacionados) e `IN`/`NOT IN` (con subconsulta) son las herramientas de SQL para expresar $\exists/\neg\exists$; combinando dos de ellos se obtiene $\forall$ (vía $\forall x.\varphi \equiv \neg\exists x.\neg\varphi$).
- Lógica de 3 valores: toda comparación (`=`,`<`,...) con `NULL` vale `unknown`; sólo `IS NULL`/ `IS NOT NULL` valen siempre `true`/`false`. Una fila satisface el `WHERE` si y sólo si la condición vale `true` (`unknown` se descarta igual que `false`).

| $\theta$ | `NOT` $\theta$ |
|---|---|
| true | false |
| false | true |
| unknown | unknown |

| $\theta_1$ | $\theta_2$ | `OR` | `AND` |
|---|---|---|---|
| true | unknown | true | unknown |
| false | unknown | unknown | false |
| unknown | unknown | unknown | unknown |

- **Cuidado con** `NOT IN`: si la subconsulta puede devolver algún `NULL`, entonces `x NOT IN (...)` vale `unknown` **para toda fila** (incluso las que uno esperaría que calificaran), y el resultado puede quedar vacío. La alternativa segura es `NOT EXISTS` correlacionado.

## Consejos generales

- Identifique primero *qué relaciones debe combinar* y con qué atributo se conectan.
- Para *"que tengan dos o más..."*: dos testigos distintos (dos $\exists$ con $\neg(\cdot = \cdot)$; dos copias renombradas con $\rho$).
- Para *"que NO tengan ningún..."*: negación de un existencial en lógica, o $\setminus$ en álgebra.
- Para *"para TODOS los..."*: use $\forall x.\varphi \equiv \neg\exists x.\neg\varphi$; en álgebra, separe un **universo de candidatos** de una relación de **evidencia**, y reste los pares candidato-objetivo que falten (patrón usado en A.1(h) de la guía y generalizado en el Ejercicio B.5(3)). En esta ayudantía el "objetivo" del $\forall$ puede ser a su vez el resultado de un $\exists$ anidado: identifique primero ese conjunto derivado antes de aplicar el patrón.
- En álgebra, al comparar dos tuplas de la misma relación, **renombre una copia con** $\rho$ antes de $\bowtie/\times$.

# 1 Parte A — Consultas en contexto (avanzado)

Para cada consulta, salvo que se indique lo contrario, entregue **(i)** una consulta de primer orden, **(ii)** una expresión de álgebra relacional, y **(iii)** la consulta SQL equivalente.

## 1.1 Contexto: Ferretería y Distribuidora "El Tornillo Feliz"

---

Una PYME chilena dedicada a la venta de artículos de ferretería registra su operación en el siguiente esquema:

$$\begin{aligned}
\textbf{Sucursal}(\mathsf{idSucursal}, \mathsf{comuna}, \mathsf{direccion}) &\quad \mathsf{ar} = 3\\
\textbf{Empleado}(\mathsf{rut}, \mathsf{nombre}, \mathsf{cargo}, \mathsf{idSucursal}) &\quad \mathsf{ar} = 4\\
\textbf{Cliente}(\mathsf{rutCliente}, \mathsf{nombre}, \mathsf{comuna}) &\quad \mathsf{ar} = 3\\
\textbf{Proveedor}(\mathsf{idProveedor}, \mathsf{nombre}, \mathsf{pais}) &\quad \mathsf{ar} = 3\\
\textbf{Producto}(\mathsf{sku}, \mathsf{nombre}, \mathsf{categoria}, \mathsf{precio}) &\quad \mathsf{ar} = 4\\
\textbf{Suministra}(\mathsf{idProveedor}, \mathsf{sku}) &\quad \mathsf{ar} = 2\\
\textbf{Pedido}(\mathsf{idPedido}, \mathsf{rutCliente}, \mathsf{rut}, \mathsf{idSucursal}, \mathsf{fecha}) &\quad \mathsf{ar} = 5\\
\textbf{DetallePedido}(\mathsf{idPedido}, \mathsf{sku}, \mathsf{cantidad}) &\quad \mathsf{ar} = 3
\end{aligned}$$

donde el atributo idSucursal de **Empleado** indica en qué sucursal trabaja el empleado *rut* (en SQL, este atributo puede ser `NULL` para un empleado recién contratado que aún no ha sido asignado a una sucursal); **Suministra** es multivaluada (un producto puede ser suministrado por varios proveedores, y un proveedor suministra varios productos); **Pedido**($idPedido, rutCliente, rut, idSucursal, fecha$) indica que el pedido *idPedido* fue realizado por el cliente *rutCliente*, atendido por el empleado *rut*, en la sucursal *idSucursal* (en SQL, *rutCliente* puede ser `NULL` para una venta de mostrador a un cliente no registrado); y **DetallePedido**($idPedido, sku, cantidad$) indica que el pedido *idPedido* incluyó *cantidad* unidades del producto *sku*. El precio de un producto (en SQL) puede ser `NULL` si aún no ha sido fijado.

*Instancia de ejemplo (opcional, para verificar sus respuestas; los valores `NULL` sólo son relevantes para las consultas en SQL):*

**Sucursal**

| idSucursal | comuna | direccion |
|---|---|---|
| 1 | Providencia | Av. Providencia 1234 |
| 2 | Maipú | Camino Rinconada 500 |
| 3 | Las Condes | Av. Apoquindo 4500 |

**Empleado**

| rut | nombre | cargo | idSucursal |
|---|---|---|---|
| 101 | Juan Pérez | Vendedor Senior | 1 |
| 102 | María Soto | Vendedor | 1 |
| 103 | Pedro Gómez | Vendedor Senior | 2 |
| 104 | Camila Ruiz | Bodeguero | `NULL` |

**Cliente**

| rutCliente | nombre | comuna |
|---|---|---|
| 201 | Ana Torres | Providencia |
| 202 | Luis Vidal | Maipú |
| 203 | Carla Muñoz | Ñuñoa |
| 204 | Diego Rojas | Providencia |

**Proveedor**

| idProveedor | nombre | pais |
|---|---|---|
| 301 | FerroChile | Chile |
| 302 | ToolMax | Estados Unidos |
| 303 | Hierros del Sur | Chile |
| 304 | NordicTools | Suecia |

**Producto**

| sku | nombre | categoria | precio |
|---|---|---|---|
| 1 | Tornillo 1" | Tornillos | 50 |
| 2 | Tornillo 2" | Tornillos | 70 |
| 3 | Pintura Blanca 1L | Pinturas | 8990 |
| 4 | Taladro Eléctrico | Herramientas | NULL |
| 5 | Combo de Brocas | Herramientas | 5990 |
| 6 | Guantes de Seguridad | Seguridad | 2990 |

**Suministra**

| idProveedor | sku |
|---|---|
| 301 | 1 |
| 301 | 3 |
| 302 | 1 |
| 302 | 4 |
| 303 | 2 |
| 303 | 5 |
| 304 | 6 |

**Pedido**

| idPedido | rutCliente | rut | idSucursal | fecha |
|---|---|---|---|---|
| 1001 | 201 | 101 | 1 | 2026-01-05 |
| 1002 | 202 | 103 | 2 | 2026-01-06 |
| 1003 | 201 | 103 | 2 | 2026-01-10 |
| 1004 | 203 | 101 | 1 | 2026-01-07 |
| 1005 | NULL | 102 | 1 | 2026-01-08 |
| 1006 | 204 | 101 | 1 | 2026-01-09 |

**DetallePedido**

| idPedido | sku | cantidad |
|---|---|---|
| 1001 | 1 | 10 |
| 1001 | 3 | 2 |
| 1002 | 2 | 5 |
| 1003 | 5 | 1 |
| 1004 | 1 | 3 |
| 1005 | 2 | 4 |
| 1006 | 3 | 1 |

---

**Ejercicio A.1.** Obtenga el nombre de los clientes que han realizado al menos un pedido en **todas** las sucursales que cuentan con algún empleado de cargo "Vendedor Senior".

**Ejercicio A.2.** Obtenga el nombre de los proveedores **extranjeros** (país distinto de "Chile") que **no** suministran ningún producto que también sea suministrado por algún proveedor chileno.

**Ejercicio A.3 (única con agregación — responda sólo en SQL).** Obtenga el identificador y la comuna de cada sucursal, junto con el monto total vendido (suma de *cantidad* $\times$ *precio* de todos los pedidos que atendió), pero **sólo** para las sucursales cuyo monto total supere el promedio de los montos totales de todas las sucursales.

## 2 Parte B — División relacional: poder expresivo y propiedades

Recuerde la definición de $\div$ dada en la Notación: para $R, S$ con $\mathsf{att}(S) \subsetneq \mathsf{att}(R)$ y $A := \mathsf{att}(R) \setminus \mathsf{att}(S)$,

$$R \div S \;:=\; \big\{\bar{\tau} \in \pi_A(R) \;\big|\; \forall \bar{\sigma} \in S.\ (\bar{\tau} \bowtie \bar{\sigma}) \in R\big\}.$$

**Ejercicio B.1 (¿÷ agrega poder expresivo?).** Antes de demostrar propiedades de $\div$, verifiquemos que este operador **no** extiende el álgebra relacional a un lenguaje más expresivo: al igual que con el semijoin y el anti-join, debemos poder simularlo usando sólo los operadores base $(\cup, \cap, \setminus, \sigma, \rho, \pi, \bowtie)$.

(a) ¿Por qué la definición exige explícitamente que $\bar{\tau} \in \pi_A(R)$, en lugar de simplemente pedir que $\bar{\tau}$ sea cualquier tupla con atributos $A$? (Piense qué pasaría si $S = \varnothing$: ¿por qué sin esa exigencia $R \div S$ podría no ser una relación válida, es decir, un conjunto **finito** de tuplas?)

(b) Demuestre que

$$R \div S \;=\; \pi_A(R) \;\setminus\; \pi_A\Big(\big(\pi_A(R) \times S\big) \setminus R\Big),$$

es decir, que $\div$ es expresable enteramente en términos de $\times, \setminus, \pi$ (por lo tanto, tan potente como el álgebra relacional vista en clases, ni más ni menos). *Hint:* la expresión $\big(\pi_A(R) \times S\big) \setminus R$ calcula, para cada candidato $\bar{\tau} \in \pi_A(R)$ y cada $\bar{\sigma} \in S$, los pares $\bar{\tau} \bowtie \bar{\sigma}$ que **no** están en $R$; proyectar esos pares sobre $A$ da exactamente los candidatos que fallan para *algún* $\bar{\sigma} \in S$.

Con esta equivalencia establecida, sean ahora $S, S'$ relaciones con $\mathsf{att}(S) = \mathsf{att}(S') \subsetneq \mathsf{att}(R)$.

**Ejercicio B.2.** Demuestre que

$$R \div (S \cup S') \;=\; (R \div S) \cap (R \div S').$$

**Ejercicio B.3.** ¿Es cierto, en general, que

$$R \div (S \cap S') \;=\; (R \div S) \cup (R \div S')\ ?$$

Demuestre que la contención $\supseteq$ siempre se cumple, y decida si la igualdad es válida en general: si lo es, demuéstrela; si no, entregue un contraejemplo concreto (relaciones $R, S, S'$ con sus tuplas).

# 3 Parte C — Nulos y equivalencia de consultas

Use la tabla **Producto** y el resto del esquema de la Parte A (incluyendo los valores `NULL` de la instancia de ejemplo).

**Ejercicio C.1.** Sea $\theta := (\mathsf{precio} > 100)$, evaluado con lógica de 3 valores sobre **Producto** (que puede tener $\mathsf{precio} = \texttt{NULL}$). ¿Es cierto que

$$\sigma_{\theta}(\textbf{Producto}) \cup \sigma_{\neg\theta}(\textbf{Producto}) \;=\; \textbf{Producto}\ ?$$

Justifique con la instancia de ejemplo (piense en el producto sin precio fijado).

**Ejercicio C.2.** Considere las siguientes dos consultas SQL, que buscan obtener el nombre de los clientes que **no** han comprado ningún producto de categoría "Tornillos":

```
-- Consulta A
SELECT c.nombre
FROM Cliente c
WHERE c.rutCliente NOT IN (
    SELECT p.rutCliente
    FROM Pedido p
    JOIN DetallePedido d ON d.idPedido = p.idPedido
    JOIN Producto pr ON pr.sku = d.sku
    WHERE pr.categoria = 'Tornillos'
);

-- Consulta B
SELECT c.nombre
FROM Cliente c
WHERE NOT EXISTS (
    SELECT 1
    FROM Pedido p
    JOIN DetallePedido d ON d.idPedido = p.idPedido
    JOIN Producto pr ON pr.sku = d.sku
    WHERE pr.categoria = 'Tornillos'
      AND p.rutCliente = c.rutCliente
);
```

¿Son equivalentes? Evalúe ambas sobre la instancia de ejemplo (note que el pedido 1005 es una venta de mostrador, con $\mathsf{rutCliente} = \texttt{NULL}$) y justifique su respuesta usando lógica de 3 valores.