

## Página

<https://dbis-uibk.github.io/relax/calc/local/uibk/local/0>

## Selección $\sigma$

Selecciona ciertas tuplas que satisfacen el predicado dado  $\sigma_{\text{predicado}}(\text{Tabla})$

- Predicado: Se trata de una operación que incluye cualquiera de los operadores  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $\wedge$ ,  $\vee$ ,  $\neg$

### Empleado

RFC	Nombre	sueldo	depto	ingreso
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004
HEFC790926	Carlos Hernández	\$ 8,000.00	A2	01/01/2003
ROPJ081001	Jorge Rojas	\$ 10,000.00	A2	01/10/2003
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002
PEVN780704	Noé Pérez	\$ 15,000.00	A3	01/06/2002
JUBF910410	Fabián Juárez	\$ 6,000.00	A2	01/11/2003
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002
RUZP810512	Patricia Ruíz	\$ 8,000.00	A2	01/04/2003
LOMM750919	María López	\$ 10,000.00	A3	01/10/2002

$\sigma_{\text{sueldo} \geq 8000 \wedge \text{ingreso} \geq 01/01/2003}(\text{Empleado})$

RFC	Nombre	sueldo	depto	ingreso
HEFC790926	Carlos Hernández	\$ 8,000.00	A2	01/01/2003
ROPJ081001	Jorge Rojas	\$ 10,000.00	A2	01/10/2003
RUZP810512	Patricia Ruíz	\$ 8,000.00	A2	01/04/2003

## Proyección $\pi$

Obtiene una tabla a partir de la tabla original eliminando los atributos no especificados. En la tabla resultante aparecen los atributos en el mismo orden que en la lista. Otra particularidad que tiene es que las tuplas duplicadas se eliminan. Su sintaxis es:

$\pi_{\text{lista\_de\_atributos}}(\text{Tabla})$

Tomando como referencia la tabla anterior de Empleado:

$\pi_{\text{sueldo, depto}}(\text{Empleado})$

sueldo	depto
\$ 6,000.00	A1
\$ 8,000.00	A2
\$ 10,000.00	A2
\$ 8,000.00	A1
\$ 12,000.00	A1
\$ 15,000.00	A3
\$ 6,000.00	A2
\$ 10,000.00	A3

A diferencia de la selección, la proyección no puede evaluar predicados, o sea que para filtrar deberíamos seleccionar para luego proyectar.

## Unión $\cup$

Obtiene una tabla que contiene las tuplas de la primera relación además de las tuplas de la segunda relación. Los renglones duplicados se eliminan.

Al adaptar los operadores de conjuntos a las relaciones se debe asegurar que exista compatibilidad entre ellas:

- **Mismo grado:** El *grado* de una relación es el número de atributos (columnas) que tiene.

Ejemplo: si la tabla Empleado tiene 5 atributos (ID, Nombre, Sueldo, Depto, Ingreso), otra relación que quieras unir con ella también debe tener 5 atributos. Si una tiene 4 columnas y la otra 5, no se pueden unir directamente con  $\cup$ .

- **Mismos nombres de atributos:** Los nombres de las columnas deben coincidir.

Ejemplo: si en Empleado la columna se llama Nombre, en la otra tabla también debe llamarse Nombre, no Nom o Empleado\_Nombre.

- **Mismos dominios de atributos:** El *dominio* es el tipo de dato permitido en esa columna.

Ejemplo: si en la primera relación la columna sueldo es un número (int o decimal), en la segunda relación no puede ser texto (varchar). En otras palabras: la columna  $i$  de  $R$  debe ser del mismo tipo de dato que la columna  $i$  de  $S$ .

Considerando la tabla Empleado :

```
r =  $\sigma_{\text{sueldo} \geq 10000}(\text{Empleado})$   
s =  $\sigma_{\text{depto} = A1}(\text{Empleado})$   
 $\pi_{\text{nombre}}(r \cup s)$ 
```

Nombre
Jorge Rojas
Germán Flores
Noé Pérez
Carlos Hernández
María López
Verónica Sánchez
Oscar Romero

## Diferencia -

Crea una tabla con las tuplas que están en la relación  $R$  pero no en  $S$ . Esta operación se da en tablas compatibles como se explica en Unión. Los renglones duplicados se eliminan.

Considerando la tabla Empleado :

$r = \sigma_{\text{sueldo} \geq 10000}(\text{Empleado})$   
 $s = \sigma_{\text{depto} = A1}(\text{Empleado})$   
 $\pi_{\text{nombre}}(r-s)$

Nombre	sueldo
Germán Flores	\$ 12,000.00
Noé Pérez	\$ 15,000.00
Carlos Hernández	\$ 12,000.00
María López	\$10,000.00

## Intersección $\cap$

Se trata de una relación con las tuplas que están en R y también en S exclusivamente, o sea que cualquier tupla que no comparta esta condición será descartada. Esta operación es válida únicamente en relaciones compatibles, como se explica en Unión. Los renglones duplicados se eliminan.

Considerando la tabla Empleado :

$r = \pi_{\text{sueldo}}(\sigma_{\text{depto}=A1}(\text{Empleado}))$   
 $s = \pi_{\text{sueldo}}(\sigma_{\text{depto}=A2}(\text{Empleado}))$   
 $r \cap s$

sueldo
\$ 6,000.00
\$ 8,000.00

## Producto cartesiano $\times$

Permite combinar información de cualquier par de relaciones R x S, dando como resultado todas las combinaciones posibles, pero es necesario renombrar si tienen atributos en común para evitar ambigüedad. Para esto generalmente se precede el nombre del atributo con el nombre de la relación.

Considerando la tabla Empleado y la tabla Departamento:

### Empleado

RFC	Nombre	sueldo	depto	ingreso
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004
HEFC790926	Carlos Hernández	\$ 8,000.00	A2	01/01/2003
ROPJ081001	Jorge Rojas	\$ 10,000.00	A2	01/10/2003
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002
PEVN780704	Noé Pérez	\$ 15,000.00	A3	01/06/2002
JUBF910410	Fabián Juárez	\$ 6,000.00	A2	01/11/2003
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002
RUZP810512	Patricia Ruíz	\$ 8,000.00	A2	01/04/2003
LOMM750919	María López	\$ 10,000.00	A3	01/10/2002

### Departamento

depto	nombre	fecha
A1	Sistemas	01/03/2002
A2	Mercadotecnia	01/01/2002
A3	Ventas	01/01/2001
A4	Recursos Humanos	01/01/2003

$\sigma_{\text{depto}=A1}(\text{Empleado}) \times \text{Departamento}$

RFC	Empleado.Nombre	sueldo	depto	ingreso	depto	departamento.nombre	fecha
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004	A1	Sistemas	01/03/2002
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002	A1	Sistemas	01/03/2002
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002	A1	Sistemas	01/03/2002
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002	A1	Sistemas	01/03/2002
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004	A2	Mercadotécnica	01/01/2002
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002	A2	Mercadotécnica	01/01/2002
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002	A2	Mercadotécnica	01/01/2002
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002	A2	Mercadotécnica	01/01/2002
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004	A3	Ventas	01/01/2001
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002	A3	Ventas	01/01/2001
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002	A3	Ventas	01/01/2001
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002	A3	Ventas	01/01/2001
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004	A4	Recursos Humanos	01/01/2003
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002	A4	Recursos Humanos	01/01/2003
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002	A4	Recursos Humanos	01/01/2003
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002	A4	Recursos Humanos	01/01/2003

Trajo todas las combinaciones posibles, en el caso de que yo quisiera solamente los departamentos correspondientes al depto, debería usar Join.

## Join Natural ⋈

El join es como el producto cartesiano pero con lógica, en lugar de quedarnos con todas las combinaciones, solo nos quedamos con las que “tienen sentido” de acuerdo a una condición. Los atributos coincidentes se superponen, es decir, no se repiten ni es necesario preceder por ambigüedad.

El join natural solo se puede usar en caso de tener igual coincidencia de atributos.

$\pi_{RFC,nombre,depto,nom\_depto}(Empleado \bowtie Departamento)$

RFC	Nombre	depto	nom_depto
SACV750524	Verónica Sánchez	A1	Sistemas
HEFC790926	Carlos Hernández	A2	Mercadotécnica
ROPJ081001	Jorge Rojas	A2	Mercadotécnica
ROAO740114	Oscar Romero	A1	Sistemas
FORG760206	Germán Flores	A1	Sistemas
PEVN780704	Noé Pérez	A3	Ventas
JUBF910410	Fabián Juárez	A2	Mercadotécnica
HEVC950826	Carlos Hernández	A1	Sistemas
RUZP810512	Patricia Ruíz	A2	Mercadotécnica
LOMM750919	María López	A3	Ventas

## Theta Join

Hace referencia a que vamos a poder darle un grado o condición a Join, es como hacer el producto cartesiano primero y luego aplicar una selección ( $\sigma$ ) con una condición. En el caso

anterior si quisiera discriminar por depto queriendome quedar solo con las tuplas de 'A1' tendria que hacer:

$\pi_{RFC,nombre,depto,nom\_depto}(Empleado \bowtie_{depto=A2} Departamento)$

RFC	Nombre	depto	nom_depto
HEFC790926	Carlos Hernández	A2	Mercadotécnica
ROPJ081001	Jorge Rojas	A2	Mercadotécnica
JUBF910410	Fabián Juárez	A2	Mercadotécnica
RUZP810512	Patricia Ruíz	A2	Mercadotécnica

### Left outer join $\bowtie_{\leftarrow}$

Quiero a todos los de la izquierda, aunque no tengan pareja en la derecha. todo A + la intersección.

- Devuelve todas las filas de la tabla izquierda.
- Si no hay coincidencia en la tabla derecha, completa con NULL.

Ejemplo:

- Todos los Clientes, aunque no hayan hecho pedidos.
- Si un cliente no tiene pedidos  $\rightarrow$  igual aparece, pero con columnas de Pedido = NULL.

### Right outer join $\bowtie_{\rightarrow}$

Quiero a todos los de la derecha, aunque no tengan pareja en la izquierda. Todo B + la intersección.

- Es lo mismo que el LEFT, pero al revés.
- Devuelve todas las filas de la tabla derecha.
- Si no hay coincidencia en la izquierda  $\rightarrow$  NULL en las columnas de esa tabla.

Ejemplo:

- Todos los **Pedidos**, aunque no tengan un cliente válido (quizá un pedido huérfano cargado mal).
- Cliente = NULL si no existe.

### Full outer join $\bowtie_{\leftrightarrow}$

Quiero a todos de ambos lados. Si no hay match, relleno con NULL. Es como hacer LEFT OUTER JOIN + RIGHT OUTER JOIN juntos. A  $\cup$  B (todo).

- Devuelve todas las filas de las dos tablas.

- Si hay coincidencia, se muestran juntas.
- Si no, aparecen con NULL en los atributos que falten.

Ejemplo:

- Todos los **Cientes** (con pedidos si tienen, NULL si no).
- Todos los **Pedidos** (con cliente si existe, NULL si no).

### Left semi join ✕

Quiero los de la izquierda que sí tienen pareja en la derecha. Solo A, pero solo los que tienen match en B (sin mostrar B).

- Parece un NATURAL JOIN, pero sólo devuelve las filas de la izquierda, sin duplicarlas ni agregar columnas de la derecha.
- Sirve para “filtrar” la izquierda con la existencia en la derecha.

Ejemplo:

- Clientes que hicieron al menos un pedido.
- Resultado: lista de clientes, sin mostrar datos de pedidos.

### Right semi join ✕

Igual que el anterior, pero al revés: Solo B, pero solo los que tienen match en A.

- "Quiero los de la derecha que sí tienen pareja en la izquierda."

Ejemplo:

- Pedidos que tienen un cliente válido.
- Devuelve la lista de pedidos, sin mostrar datos de clientes.

### Anti join ▷

Quiero a los que no tienen pareja. Los que se quedan fuera del match (parte de A sin B, o de B sin A).

- LEFT ANTI JOIN: los de la izquierda que NO tienen match en la derecha.  
Ejemplo: clientes que nunca hicieron pedidos.
- RIGHT ANTI JOIN: los de la derecha que NO tienen match en la izquierda.
- Ejemplo: pedidos que no corresponden a ningún cliente (error de datos).



## Renombramiento $\rho$

Asigna un nombre a la relación y/o atributos, sería como el AS en SQL.

- Renombrar una relación:  $\rho_{\text{NuevoNombre}}(\text{Relación})$   
Ejemplo: Si tengo la relación Empleado(RFC, Nombre, Sueldo, Depto, Ingreso), y quiero renombrarla a Trabajador se haría de la siguiente forma:  
 $\rho_{\text{Trabajador}}(\text{Empleado})$ .
- Renombrar un atributo:  $\rho_{\text{Tabla}}(A_1, A_2, \dots, A_n)(\text{Relación})$   
Ejemplo: Si quiero que RFC se llame ID y Sueldo se llame Salario:  
 $\rho_{\text{Trabajador}}(\text{ID}, \text{Nombre}, \text{Salario}, \text{Depto}, \text{Ingreso})(\text{Empleado})$

Es necesario renombrar cuando, por ejemplo, quiero hacer un **producto cartesiano** de Empleado consigo misma (para ver, por ejemplo, quién gana más que quién). Si no renombro, los atributos chocan porque tienen el mismo nombre en las dos copias. Entonces hago:  $\rho_{E1}(\text{Empleado}) \times \rho_{E2}(\text{Empleado})$ . Ahora puedo distinguir entre  $E1.\text{Nombre}$ ,  $E2.\text{Nombre}$ ,  $E1.\text{Sueldo}$ ,  $E2.\text{Sueldo}$ .

## Agrupación $\gamma$

Permite agrupar conjuntos de valores en función de un atributo determinado y hacer operaciones con otros campos, es decir, nos permite juntar tuplas en grupos que comparten un mismo valor en ciertos atributos, y sobre esos grupos aplicar funciones de agregación.

- SUM: Suma
- AVG(Atributo): Promedio
- COUNT: Contar
- MIN: Mínimo
- MAX: Máximo

Se escribe como  $\gamma_{\text{atributos}}; \text{funcion1}, \text{funcion2}, \dots (\text{Relación})$

Ejemplo: Si quiero el sueldo promedio por departamento

$\gamma_{\text{Depto}}; \text{AVG}(\text{Sueldo}) \rightarrow \text{sueldoprom}(\text{Empleado})$

depto	sueldoprom
A1	\$ 9,500.00
A2	\$ 8,000.00
A3	\$ 12,500.00

(El operador  $\rightarrow$  renombro la columna temporal para el promedio)

## Ordenamiento $\tau$

El ordenamiento organiza las tuplas de una relación en un orden específico según uno o varios atributos. Este orden puede ser asc (ascendente) o desc (descendente), pero si no se especifica siempre va a ser ascendente por omisión. El operador  $\tau$  solo va a tener sentido cuando se usa como el operador final de una expresión.

Se escribe como  $\tau_{\text{atributo}} [\text{ASC/DESC}] (\text{Relación})$

Ejemplo: empleados ordenados por sueldo (de mayor a menor)  
 $\tau_{\text{sueldo DESC}} (\text{Empleado})$

## División $\div$

Busca tuplas que cumplen una relación completa con otra tabla, es decir, que crea una nueva tabla seleccionando las filas en una relación que se corresponden con todas las filas en otra relación. Ejemplo y sintaxis:

Dado la tabla Empleado:

$S = \sigma_{\text{sueldo} \geq 6000} (\text{Empleado} \bowtie \text{Departamento})$   
 $R = \pi_{\text{depto, sueldo}} (S)$

r	
depto	sueldo
A1	\$ 6,000.00
A2	\$ 6,000.00

Si divido:

$\text{Empleado} \div R$

RFC	Nombre	ingreso
SACV750524	Verónica Sánchez	01/01/2004
JUBF910410	Fabián Juárez	01/11/2003

Básicamente me devolvió aquellos datos relacionados con R de la tabla Empleado que no están en R.

## Mantenimiento de datos

- **Borrado**

El borrado es equivalente a  $R = R - \sigma_{\text{cond}}(R)$ , y tiene como restricción tener el mismo grado y los dominios adecuados.

Ejemplo de la tabla Empleado: Necesito borrar todos los empleados del departamento A2.



$\text{Empleado} = \text{Empleado} - \sigma_{\text{Depto} = 'A2'}(\text{Empleado})$

- **Inserción**

La inserción tiene esta forma  $R = R \cup (\text{resultadoConsulta})$ , y tiene la misma restricción que el borrado.

Ejemplo de la tabla Empleado:

- **Actualización**

La actualización funciona usando el borrado y luego la inserción, ya que se puede modelar como

1. Quitar las tuplas antiguas que cumplen la condición.
2. Insertar las **nuevas** tuplas con los valores modificados.

Ejemplo: subir 10% el sueldo de empleados de A1

1. Seleccionás las tuplas que vas a actualizar:  
 $R_{\text{viejo}} = \sigma_{\text{Depto} = 'A1'}(\text{Empleado})$
2. Construí las nuevas tuplas aplicando la proyección generalizada (modificando Sueldo):  
 $R_{\text{nuevo}} = \pi_{\text{RFC}, \text{Nombre}, \text{Sueldo1} = \text{Sueldo} * 1.10, \text{Depto}, \text{Ingreso}}(R_{\text{viejo}})$
3. Actualizás la relación:  
 $\text{Empleado} = (\text{Empleado} - R_{\text{viejo}}) \cup R_{\text{nuevo}}$