

# Álgebra Relacional

Carlos A. Olarte (carlosolarte@puj.edu.co)  
BDI

1 Operaciones Fundamentales

2 Operaciones Adicionales

Conjunto de operaciones que toman como argumentos relaciones y retornan una nueva relación.

Las operaciones fundamentales del Álgebra relacional son:

- Selección  $\sigma$
- Proyección  $\pi$
- Unión  $\cup$
- Diferencia  $-$
- Producto cartesiano  $\times$
- Renombramiento  $\rho$

Selección de tuplas dado un predicado lógico.

Por ejemplo,  $\sigma_{valor > 10 \wedge codigo < 500}(Productos)$  selecciona los productos cuyo *valor* es mayor a 10 y su *codigo* menor a 500. En SQL:

```
SELECT * FROM PRODUCTOS
WHERE VALOR > 10
      AND CODIGO < 500;
```

Selección de uno o varios atributos de una relación.

Por ejemplo  $\pi_{codigo}(Productos)$  retorna todas las tuplas de *productos* pero solo con el atributo *codigo*. En SQL:

```
SELECT CODIGO  
FROM PRODUCTOS;
```

- Composición: Combinación de dos o mas operadores del Álgebra relacional.  
Por ejemplo  $\pi_{codigo} \sigma_{valor > 10 \wedge codigo < 500} (Productos)$
- Expresión de A.R: Composición de varias operaciones de A.R

Unión de *conjuntos* de tuplas.

Por ejemplo  $\pi_{cc,saldo}(ctas\_corriente) \cup \pi_{cc,saldo}(ctas\_ahorro)$  retornaría el *conjunto*<sup>1</sup> de los saldos y cédulas de las cuentas de ahorros y de crédito. En SQL:

```
SELECT CODIGO FROM PRODUCTOS
UNION
SELECT CODIGO FROM CLIENTES;
```

---

<sup>1</sup>sin repetición

# Diferencia de Conjuntos (—)

Diferencia de *Conjuntos* de tuplas.

Por ejemplo

$\pi_{codigo}(Estudiantes) - \pi_{est\_codigo} \sigma_{mat\_codigo='CC080'}(Matriculas)$

selecciona los estudiantes que no han matriculado la materia CC080. En SQL

```
SELECT CODIGO FROM ESTUDIANTES  
MINUS  
SELECT EST_CODIGO FROM MATRICULAS  
WHERE MAT_CODIGO = 'CC080';
```



# Producto Cartesiano ( $\times$ )

Producto cartesiano entre *Conjuntos* de tuplas.

Por ejemplo  $Estudiantes \times Cursos$  retornaría la combinación entre cada estudiante en *Estudiantes* con todos los cursos en la relación *Cursos*. En SQL:

```
SELECT *  
FROM ESTUDIANTES, CURSOS
```

# Renombramiento ( $\rho$ )

Cambio del nombre de una relación y/o sus atributos.

Por ejemplo  $\rho_{emp(c,n,t)}(\pi_{cod,nom, tel} Empleado)$  retorna una relación con nombre *emp* y atributos  $(c, n, t)$ . En SQL se hace uso de alias para implementar el renombramiento, por ejemplo:

```
SELECT COD C, NOM N, TEL T  
FROM EMPLEADOS EMP;
```

A partir de las operaciones fundamentales del Álgebra relacional, es posible generar nuevas que aunque no aumentan poder de expresión si facilitan la escritura de consultas habituales.

Entre estas operaciones están: Intersección, Reunión natural, zeta y externa, división, etc.

# Intersección de Conjuntos ( $\cap$ )

Intersección entre *conjuntos* de tuplas.

$$r \cap s \equiv r - (r - s)$$

Producto cartesiano pero existe el criterio de reunión dado por los atributos comunes

$$r \bowtie s = \pi_{R \cup S}(\sigma_{r.a1=s.a1 \dots r.an=s.an} r \times s)$$

Producto cartesiano pero existe un criterio de reunión explícito  $\theta$

$$r \bowtie_{\theta} s = \sigma_{\theta}(r \times s)$$

Por ejemplo,

$\pi_{p.nombre, c.nombre}(\rho_p(Profesor) \bowtie_{p.codigo=c.pro\_codigo} \rho_c(Curso))$   
retorna una relación que contiene el nombre de cada profesor y sus respectivos cursos a cargo. En SQL:

```
SELECT P.NOMBRE, C.NOMBRE  
FROM PROFESORES P,  
      CURSOS C  
WHERE P.CODIGO = C.PRO_CODIGO;
```

## Reunión externa ( $=\bowtie$ ) -Derecha o izquierda-

Realiza el join pero si no hay match genera un valor nulo. Es útil cuando el atributo FK puede ser nulo. Por ejemplo, si no todos los cursos del ejemplo anterior tienen un profesor asignado, se debe utilizar la siguiente expresión:

$\pi_{nombre}(Profesor) \bowtie_{P.codigo=C.Pro\_codigo} \pi_{nombre}(Curso)$  En SQL :

```
SELECT P.NOMBRE, C.NOMBRE  
FROM PROFESORES P,  
CURSOS C  
WHERE C.PRO_CODIGO = P.CODIGO(+);
```

Asignación como en un lenguaje de programación:

$$\begin{aligned}r_1 &\leftarrow \pi_{\text{codigo}, \text{nombre}}(\text{Profesores}) \\r_2 &\leftarrow \pi_{\text{nombre}, \text{pro\_codigo}}(\text{Cursos}) \\r_3 &\leftarrow r_1 \bowtie_{r_1.\text{codigo}=r_2.\text{pro\_codigo}} r_2 \\r_4 &\leftarrow \pi_{r_1.\text{nombre}, r_2.\text{nombre}}(r_3)\end{aligned}$$



# División ( $r \div s$ )

Está en el esquema  $R - S$  y son las tuplas de  $r$  que están relacionadas con todas las tuplas de  $s$ .

Formalmente, sea  $r(R)$  y  $s(S)$  relaciones y  $S \subseteq R$ ,  $r \div s$  es una relación del esquema  $R - S$  y  $t \in r \div s$  si:

- ❶  $t \in \pi_{R-S}(r)$
- ❷ Para toda  $t_s$  de  $s$  hay una tupla  $t_r$  de  $r$  tal que:
  - ❶  $t_r[S] = t_s[S]$
  - ❷  $t_r[R - S] = t$

**Ejercicio:** Escribir la consulta SQL para  $r(R) \div s(S)$

Son funciones que toman un conjunto de valores y retornan un valor único según un criterio de agrupamiento. Condiciones:

- 1 Todas las tuplas del grupo tienen los mismos valores para  $g_1..g_n$
- 2 Las tuplas de diferentes grupos tienen valores distintos para  $g_1..g_n$

Ejemplo:  $mat\_codigo, periodo \ G \ count(mat\_codigo)(Matricula)$  cuenta por cada código de materia y período los alumnos matriculados. En SQL:

```
SELECT MAT_CODIGO, PERIODO,  
COUNT(MAT_CODIGO)  
FROM MATRICULAS  
GROUP BY MAT_CODIGO;
```

Para contar el total de alumnos de la universidad:

$G_{count(codigo)}(Estudiantes)$ . En SQL

```
SELECT COUNT(CODIGO)
FROM ESTUDIANTES;
```

Para calcular el promedio de notas de cada grupo por cada período:  $mat\_codigo, periodo G_{avg(nota)}(Matricula)$ . En SQL:

```
SELECT MAT_CODIGO, PERIODO,
AVG(NOTA) PROMEDIO
FROM MATRICULAS
GROUP BY MAT_CODIGO, PERIODO;
```

Dado el siguiente esquema de B.D:

*Producto*(*cod*, *nombre*, *precio*)

*Ciudad*(*cod*, *nombre*)

*Cliente*(*cc*, *Nombre*, *ciu\_cod*)

*Sucursal*(*cod*, *nombre*, *ciu\_cod*)

*Factura*(*Num*, *suc\_cod*, *Fecha*, *Cli\_Cod*)

*Detalle*(*Fac\_Num*, *Fac\_Suc\_Cod*, *Pro\_Cod*, *Cant*, *Unit*)

Escribir las siguientes consultas en A.R

- 1 Seleccionar los Números de factura con su respectivo nombre de sucursal
- 2 Seleccionar los clientes que no han sido registrados en ninguna factura
- 3 Seleccionar todos los clientes junto con sus facturas (si las tiene)
- 4 Seleccionar el total de facturas del mes  $X$
- 5 Seleccionar el total de facturas por sucursal del mes  $X$
- 6 Seleccionar el producto mas vendido
- 7 Seleccionar el producto mas vendido por sucursal
- 8 Encontrar el o los clientes que han comprado en todas las sucursales
- 9 Encontrar los clientes que solo han comprado en la sucursal  $X$