Algebra relacional

Rodrigo Barrera

Recursos

- RelaX - relational algebra calculator (dbis-uibk.github.io)

¿Por qué estudiar al álgebra relacional?

- El álgebra relacional proporciona la base teórica para las operaciones en bases de datos relacionales, que son el tipo más común de bases de datos utilizadas en aplicaciones comerciales y de software.
- Permite describir con precisión las operaciones de consulta y manipulación de datos sin referirse a cómo se ejecutarán las operaciones. Esto ayuda a comprender qué se quiere lograr con una consulta sin preocuparse por los aspectos de implementación.

¿Por qué estudiar al álgebra relacional?

Comprender el álgebra relacional es crucial para el análisis y la optimización de consultas. Permite reformular consultas de manera que se puedan ejecutar de manera más eficiente.

Aunque el SQL no es una implementación directa del álgebra relacional, muchos de sus conceptos se derivan de él. Entender el álgebra relacional ayuda a escribir consultas SQL más efectivas y a comprender mejor cómo los sistemas de gestión de bases de datos (SGBD) procesan estas consultas.

¿Qué es el modelo relacional?

- Representación lógica de datos
  - Tablas bidimensionales (relaciones)
- Sistema formal de manipulación de relaciones
  - Álgebra relacional
- Resultado
  - Descripción de alto nivel (lógica, declarativa) de los datos
  - Reglas mecánicas para reescribir/optimizar el acceso de bajo nivel
  - Métodos formales para razonar sobre la solidez

Relaciones y tuplas

Tupla ➔

R	a_1	a_2		a_m
t_1				
t_n				$v_{n,m}$

Conjunto de atributos.
$m = |schema(R)|$

¿Qué es un algebra?

- Operandos
  - Variables, constantes
  - Dominio cerrado
- Operadores
  - +
  - *
- Expresiones
  - Combinar operaciones con paréntesis (explícito)
  - O usando cualquiera de las dos precedencias (implícita)
- Leyes
  - Identifica semánticamente expresiones equivalentes
  - Conmutatividad, asociatividad, etc.

Ejemplo álgebra: aritmética entera

- Dominio: enteros
  - …-100,… -1, 0, 1, … 100, … "-" operador unario
- Operadores: -,+,*…
- Expresiones
  - 3*b – 5*c + 1
- Leyes
  - a*b=b*a
  - a(b*c) = (a*b)*c
  - a*(b+c) = a*b + a*c

Permite a los compiladores razonar y optimizar

Álgebra relacional

Valores

- Relaciones finitas
- Los atributos pueden o no estar escritos

Operadores

- Unarios: σ, π, ρ
- Aditivos (set): U, ∩, -
- Multiplicativos: ×
- [detalles por venir]

Expresiones

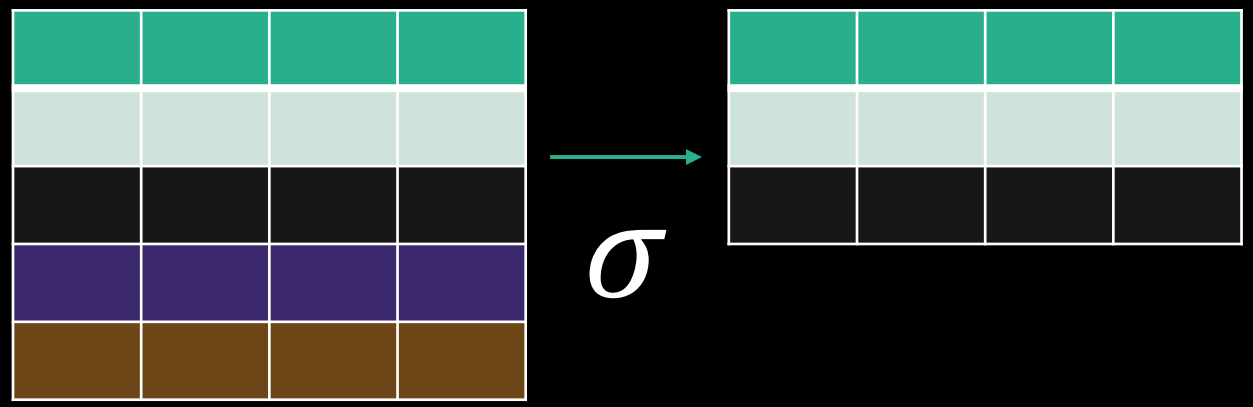
- Igual que la aritmética, pero llamada “consultas”

Leyes

- Permite reescribir consultas
- Bases para la optimización de consultas
- [detalles por venir]

Operador unario: selección (σ)

$\sigma_P(R)$ genera tuplas de R que satisfacen P
El esquema es el mismo que R



Operador unario: selección (σ)

$Employees$

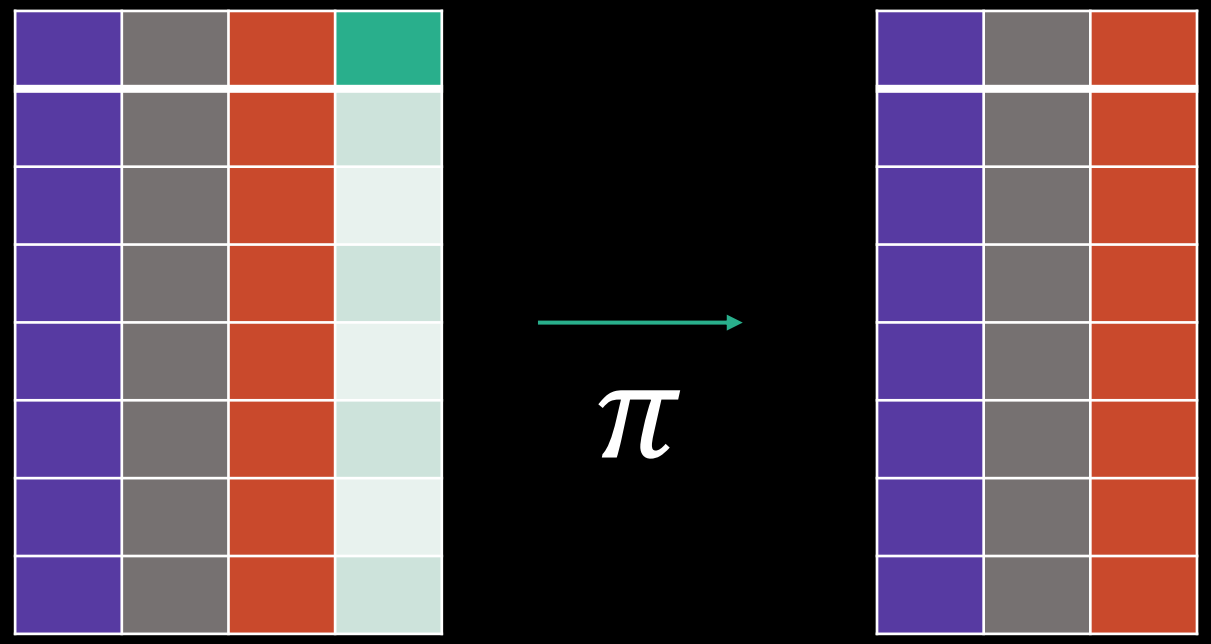
Apellido	Nombre	Edad	Salario
…	…	24	2000
…	…	40	3000
…	…	36	4500
…	…	40	3900

$\sigma_{Edad<30 \vee Salario>4000}(Employees)$

Apellido	Nombre	Edad	Salario
…	…	24	2000
…	…	36	4500

Operador unario: proyección (π)

$\pi_Y(R)$ genera un subconjunto Y del conjunto de atributos X de la relación R



Remueve columnas desde una relación

Proyección

Ejemplo

Surname	FirstName	Department	Head
Smith	Mary	Sales	De Rossi
Black	Lucy	Sales	De Rossi
Verdi	Mary	Personnel	Fox
Smith	Mark	Personnel	Fox

$\pi_{\text{Surname,FirstName}}(\text{Employees})$

Department	Head
Smith	Mary
Black	Lucy
Verdi	Mary
Smith	Mark

$\pi_{\text{Deparment, Head}}(\text{Employees})$

Department	Head
Sales	De Rossi
Personnel	Fox

Operador unario renombrar ρ

$\rho_{S(A,B,C)}(R)$ cambia el nombre de los atributos de R a A,B,C y llama al resultado S.

$\rho_S(R)$ cambia el nombre de la relación R (mismos atributos).

$\rho_{A=X,C=Y}(R)$ o $\rho_{A,C\rightarrow X,Y}(R)$ cambia el nombre de los atributos A y C solamente.

Operadores aditivos: unión (∪)

Graduates		
Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Managers		
Number	Surname	Age
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Graduates ∪ Managers

Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38
9297	O'Malley	56

Operadores aditivos: intersección (∩)

Graduates		
Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Managers		
Number	Surname	Age
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Graduates ∩ Managers		
Number	Surname	Age
7432	O'Malley	39
9824	Darkes	38

Operadores aditivos: diferencia (–)

Graduates		
Number	Surname	Age
7274	Robinson	37
7432	O'Malley	39
9824	Darkes	38

Managers		
Number	Surname	Age
9297	O'Malley	56
7432	O'Malley	39
9824	Darkes	38

Graduates - Managers		
Number	Surname	Age
7432	Robinson	37

Union con renombrar

Paternity	
Father	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael

Maternity	
Mother	
Mother	Child
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

$$P_{\{Father \rightarrow Parent\}}(Paternity) \cup P_{\{Mother \rightarrow Parent\}}(Maternity)$$

Parent	Child
Adam	Cain
Adam	Abel
Abraham	Isaac
Abraham	Ishmael
Eve	Cain
Eve	Seth
Sarah	Isaac
Hagar	Ishmael

Producto cartesiano ×

- El resultado de combinar cada registro en R con cada registro en S
- T = R × S contiene todas las combinaciones por pares de tuplas R y S.
  - Schema(T) = Schema(R) ∪ Schema(S)
  - $|T| = |R| * |S|$

Producto cartesiano ×

Employees	
Employee	Project
Smith	A
Black	A
Black	B

Projects	
Code	Name
A	Venus
B	Mars

Employee	**Project**	**Code**	**Name**
Smith	A	A	Venus
Black	A	A	Venus
Black	B	B	Mars
Smith	A	A	Venus
Black	A	A	Venus
Black	B	B	Mars

División(/)

Sea R y S relaciones con los esquemas A1, ..., An, B1, ..., Bn y B1, ..., Bn respectivamente. El resultado de R/S es una relación T con

- Esquema A1, ..., An (nombres de atributos en R pero no en S)
- Tuplas tales que, para cada tupla s de S, la tupla t|| s (la concatenación de t y s) está en relación R
- T contiene el mayor conjunto posible de tuplas s. t. $S \times T \subseteq R$

Analogía con la division de enteros

- Para enteros, A / B es: el int más grande Q t.q. Q x B ≤ A
- Para las relaciones, A / B es: la relación más grande Q t.1. $Q \times B \subseteq A$

División

Ejemplo

R

A	B	C
a1	b1	c1
a2	b1	c1
a1	b2	c1
a1	b2	c2
a2	b1	c2
a1	b2	c3
a1	b2	c4
a1	b1	c5

S1

C
c1

S2

C
c1
c2

S3

C
c1
c2
c3
c4

R/S1

A	B
a1	b1
a2	b1
a1	b2

R/S2

A	B
a1	b2
a2	b1

R/S3

A	B
a1	b2

Joins

- El operador más utilizado en álgebra relacional
- Se utiliza para establecer conexiones entre datos en diferentes relaciones, aprovechando la naturaleza ‘basada en valores’ de el modelo relacional
- Dos versiones principales del ‘join’:
  - *natural join*: tiene en cuenta los nombres de los atributos
  - *theta join*: tiene en cuenta los valores de los atributos
  - Ambas operaciones de unión denotadas por el símbolo ⋈

Natural Join ⋈

- T = R ⋈ S combina tuplas de R y S que tienen valores iguales donde sus esquemas se superponen (atributos de unión)
  - Esquema T: Unión de esquemas
  - $|T| \leq |R| * |S|$, generalmente $\approx \max(|R|, |S|)$

Casos especiales

- Sin superposición de esquema
- Superposición total de esquema

Equivalente a $\pi(\sigma(R \times \rho(S)))$

Natural Join ⋈

Ejemplo

r1	
Employee	Department
Smith	sales
Black	production
White	production

r2	
Department	Head
production	Mori
sales	Brown

r1 ⋈ r2		
Employee	Department	Head
Smith	sales	Brown
Black	production	Mori
White	production	Mori

Propiedades del ‘natural join’

- Conmutativo:

  $R \bowtie S = S \bowtie R$

- Asociativo:

  $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$

- N-ary joins sin ambigüedad:

  $R1 \bowtie R2 \bowtie \ldots \bowtie Rn$

Operador ‘natural join’

r1	
Employee	Department
Smith	sales
Black	production
Brown	marketing
White	production

r2	
Department	Division
production	A
sales	B
purchasing	B

r3	
Division	Head
A	Mori
B	Brown

r1⋈r2⋈r3			
Employee	Department	Division	Head
Black	production	A	Mori
Brown	marketing	B	Brown
White	production	A	Mori