

4. Algebra relacional

4.1 Introducción

En las secciones anteriores se han estudiado las maneras de modelar información de manera "relacional" empleando el concepto de entidades que se relacionan entre sí.

Esta sección presenta la manera de hacer consultas a una base de datos empleando algunos conceptos matemáticos aplicados a un esquema relacional. Los lenguajes que se analizan más adelante se derivan precisamente del álgebra relacional.

El álgebra relacional consiste de algunas simples pero poderosas maneras de construir nuevas relaciones a partir de otras. Si pensamos que las relaciones iniciales son los datos almacenados entonces las nuevas relaciones se pueden ver como respuestas a algunas consultas deseadas.

4.2 Conjunto de operaciones en relaciones

- $R \cup S$, la unión de R y S es el conjunto de elementos que están en R o S o ambos. Un elemento solo aparece una sola vez.
- $R \cap S$, el conjunto de elementos que aparecen en ambos R y S
- $R - S$, la diferencia de R y S , el conjunto de elementos que están en R pero no en S . Es importante resaltar que $R - S$ es diferente a $S - R$.
- R / S , la división de una relación entre otra, debe cumplirse que para toda tupla en R exista su correspondiente en S .

Restricciones:

1. R y S deben tener esquemas idénticos.
2. El orden de las columnas debe ser el mismo

Ejemplos:

name	address	gender	birthdate
Carrie Fisher	123 Maple St.	F	9/9/99
Mark Hamill	456 Oak Rd.	M	8/8/88

name	address	gender	birthdate
Harrison Ford	789 Palm Dr.	M	7/7/77
Carrie Fisher	123 Maple St.	F	9/9/99

Unión

name	address	gender	birthdate
Harrison Ford	789 Palm Dr.	M	7/7/77
Mark Hamill	456 Oak Rd.	M	8/8/88
Carrie Fisher	123 Maple St.	F	9/9/99

Intersección

name	address	gender	birthdate
Carrie Fisher	123 Maple St.	F	9/9/99

Resta

name	address	gender	birthdate
Mark Hamill	456 Oak Rd.	M	8/8/88

4.3 Proyección Π

- Crea una nueva relación a partir de otra, pero incluyendo sólo algunas de las columnas
- $\Pi_{A1,A3,A6}(R)$

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Paramount

Movie

Ejemplo:

$\Pi_{\text{title,year,length}}(\text{Movie})$

title	year	length
Star Wars	1977	124
Mighty Ducks	1991	104
Wayne's World	1992	95

$\Pi_{\text{filmType}}(\text{Movie})$

filmType
color

4.4 Selección σ

- Crea una nueva relación a partir de otra, pero incluyendo sólo algunas de las tuplas a partir de un criterio dado.
- El criterio se basa en restricciones sobre los atributos de la relación R y no pueden incluirse otras relaciones en dicho criterio que no estén en R
- $\sigma_{A3 > 16}(R)$, $\sigma_{A3 > 16 \text{ and } A3 < 45}(R)$, $\sigma_{\text{nombre}='Carlos' \text{ and } \text{edad}=45}(R)$

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Paramount

Movie

Ejemplos:

 $\sigma_{\text{length} \geq 100}$ (Movie)

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney

 $\sigma_{\text{length} \geq 100 \text{ and studioName} = \text{'Fox'}}$ (Movie)

title	year	length	filmType	studioName
Star Wars	1977	124	color	Fox

 $\pi_{\text{title, studioName}}(\sigma_{\text{length} \geq 100}(\text{Movie}))$

title	studioName
Star Wars	Fox
Mighty Ducks	Disney

4.5 Asignación <--

Almacena temporalmente el resultado de un operación en un relación dada

LOLO <-- $\pi_{\text{title, studioName}}(\sigma_{\text{length} \geq 100}(\text{Movie}))$

4.6 División

Sean

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

R

C	D
c	d
e	f

S

A	B
a	b
e	d

R / S

Ejemplo: Estudiantes que han tomado todos los cursos de "IS"

$$\pi_{ID, num} (\sigma_{depto='IS'}(estudiante_cursos)) / \pi_{num} (\sigma_{depto='IS'}(cursos))$$

4.7 Producto cartesiano X

Producto cruz o solo producto

R X S, los esquemas de ambas relaciones se mezclan y unen.

Dados

A	B
1	2
3	4

R

B	C	D
2	5	6
4	7	8
9	10	11

S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R X S

4.8 Producto natural |X|

Es un producto cartesiano donde nos interesan únicamente algunas tuplas que hacen "match" en algún criterio.

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

A	B	C	D
1	2	5	6
3	4	7	8

R |X| S

4.9 Outer Join

El outer join es una extensión del join para lidiar con información no existente. Existen 3 tipos, izquierdo, derecho y completo.

employee-name	street	city
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

employee

employee-name	branch-name	salary
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

ft-works

employee-name	street	city	branch-name	salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500

|X|

4.9.1 Left Outer Join ^{]X|}

employee-name	street	city	branch-name	salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death Valley	null	null

]X|

4.9.2 Right Outer Join ^{|X[}

employee-name	street	city	branch-name	salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Gates	null	null	Redmond	5300

|X[

4.9.3 Full Outer Join ^{]X[}

employee-name	street	city	branch-name	salary
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death Valley	null	null
Gates	null	null	Redmond	5300

]X[

4.10 Combinación de operaciones

Cuáles son los títulos y años de las películas hechas por Fox y que tengan al menos 100 minutos de duración.

1. Seleccionar aquellas películas que tienen $\text{length} \geq 100$

$$\sigma_{\text{length} \geq 100} (\text{Movie})$$

2. Seleccionar aquellas películas que tienen $\text{studioName} = \text{'Fox'}$

$$\sigma_{\text{studioName} = \text{'Fox'}} (\text{Movie})$$

3. Calcular la intersección de (1) y (2)

$$\sigma_{\text{length} \geq 100} \cap \sigma_{\text{studioName} = \text{'Fox'}} (\text{Movie})$$

4. Proyectar únicamente title y year

$$\pi_{\text{title}, \text{studioName}} (\sigma_{\text{length} \geq 100} \cap \sigma_{\text{studioName} = \text{'Fox'}} (\text{Movie}))$$

Problema:

Dadas las 2 relaciones siguientes, indique un query en algebra relacional para encontrar los nombres de las estrellas que trabajan en películas cuya duración sera mayor o igual que 100.

Movie (title,year,length,filmType,studioName)

Movie_star(title,year,starName)

$$\pi_{\text{starName}} (\sigma_{\text{length} \geq 100} (\text{Movie} \bowtie \text{Movie_star}))$$

4.11 Renombramiento ρ

Renombrar una relación para facilitar la interacción con otras

$$\rho_s (R)$$

Ej.

$$\pi_{t.\text{nombre}} (\sigma_{s.\text{nombre} = \text{'carlos'}} \text{ and } t.\text{curso} = \text{'IS341'} (\rho_s (\text{PROFE}) \bowtie \rho_t (\text{CURSO})))$$

Renombrar un atributo

Suponiendo $R (A,B,C)$

$\rho_{R(A,X,C)} (R)$	$\rho_{S(A,X,C)} (R)$	$\pi_{A, B \text{ as } X, C} (R)$	$\pi_{B \text{ as } X} (R)$
$= R(A,X,C)$	$= S(A,X,C)$	$= R(A,X,C)$	$= R(X)$

4.12 Modificaciones a la base de datos

4.12.1 Eliminación

$$r \leftarrow r - E$$

$\text{depositor} \leftarrow \text{depositor} - \sigma_{\text{customer-name}='Smith'}(\text{depositor})$

4.12.2 Inserción

$r \leftarrow r \cup E$

$\text{account} \leftarrow \text{account} \cup \{(A-973, \text{"Perryridge"}, 1200)\}$

4.12.3 Actualización

$r \leftarrow \pi_{F_1, F_2, \dots, F_n}(r)$

$\text{account} \leftarrow \pi_{\text{account-number}, \text{branch-name}, \text{balance} * 1.05}(\text{account})$

Si sólo queremos actualizar algunas tuplas:

$r \leftarrow \pi_{F_1, F_2, \dots, F_n}(\sigma_P(r)) \cup (r - \sigma_P(r))$

Suponiendo que se desea que las cuentas con balance superior a \$ 10,000 reciban un aumento del 6% y que todas las demas solo el 5%

$\text{account} \leftarrow \pi_{AN, BN, \text{balance} * 1.06}(\sigma_{\text{balance} > 10000}(\text{account})) \cup \pi_{AN, BN, \text{balance} * 1.05}(\sigma_{\text{balance} \leq 10000}(\text{account}))$

4.13 Operaciones dependientes e independientes

$$R \cap S = R - (R - S)$$

$$R \bowtie S = \pi_L(\sigma_C(R \times S))$$

4.14 Operadores Extendidos

No son parte del estándar del Algebra Relacional, pero al ser incluidos en los lenguajes de consulta más populares se han introducido como una extensión.

4.14.1 Eliminación de duplicidad

A	B
1	2
3	4
1	2
1	2

$\delta(R)$

A	B
1	2
3	4

4.14.2 Operadores de agregación

A	B
1	2
3	4
1	2
1	2

- SUM(B)= 10
- AVG(A)= 1.5
- MIN(A)=1
- MAX(B)=4
- COUNT(A)=4

Es importante resaltar que estos operadores nunca devuelven un "valor" sino una relación conteniendo el valor.

SUM(B)
10

$\pi_{\text{SUM(B)}}(R)$

4.14.3 Agrupación γ

A	B
1	2
3	4
1	2
2	8
1	2
2	6

A	SUM(B)
1	6
3	4
2	14

$\gamma_{A, \text{SUM(B)}}(R)$

4.14.4 Ordenamiento τ

$\tau_{A4, A5}(R)$

A	SUM(B)
1	6
2	14
3	4

$\tau_A(\gamma_{A, \text{SUM(B)}}(R))$

4.15 Otros lenguajes de consulta

4.15.1 Cálculo relacional de tuplas

account-number	branch-name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

account

branch-name	branch-city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

branch

customer-name	customer-street	customer-city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto

customer

customer-name	account-number
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

depositor

loan-number	branch-name	amount
L-11	RoundHill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500
null	null	1900

loan

customer-name	loan-number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17
Johnson	null

borrower

- Encontrar branch-name, loan-number and amount para los préstamos superiores a \$1,200

$$\{ t \mid t \in \text{loan} \wedge t[\text{amount}] > 1200 \}$$

- Encontrar el loan-number de cada préstamo cuyo monto sea mayor a \$1,200

$$\{ t \mid \exists s \in \text{loan} (t[\text{loan-number}] = s[\text{loan-number}] \wedge s[\text{amount}] > 1200) \}$$

- Encontrar los nombres de los clientes que tienen un préstamos de la sucursal Perryridge

$$\{ t \mid \exists s \in \text{borrower} (t[\text{customer-name}] = s[\text{customer-name}] \wedge \exists u \in \text{loan} (u[\text{loan-number}] = s[\text{loan-number}] \wedge u[\text{branch-name}] = \text{"Perryridge"})) \}$$

- Encontrar todos los clientes que tienen un préstamo, cuenta, o ambos en el banco.

$$\{ t \mid \exists s \in \text{borrower} (t[\text{customer-name}] = s[\text{customer-name}]) \\ \vee \exists u \in \text{depositor} (t[\text{customer-name}] = u[\text{customer-name}]) \}$$

- Encontrar todos los clientes que tienen una cuenta en el banco pero no tienen ningún préstamo.

$$\{ t \mid \exists u \in \text{depositor} (t[\text{customer-name}] = u[\text{customer-name}]) \wedge \\ \neg \exists s \in \text{borrower} (t[\text{customer-name}] = s[\text{customer-name}]) \}$$

- También es posible usar el símbolo "para todo" \forall junto con una implicación $P \rightarrow Q$ donde si P es verdadero, Q también deberá serlo.

Encontrar todos los clientes que tienen una cuenta en todas las sucursales localizadas en Brooklyn

$$\{ t \mid \exists r \in \text{customer} (r[\text{customer-name}] = t[\text{customer-name}]) \wedge \\ (\forall u \in \text{branch} (u[\text{branch-city}] = \text{"Brooklyn"} \rightarrow \\ \exists s \in \text{depositor} (t[\text{customer-name}] = u[\text{customer-name}]) \wedge \\ \exists w \in \text{account} (w[\text{account-number}] = s[\text{account-number}] \wedge \\ w[\text{branch-name}] = u[\text{branch-name}]))) \}$$

- Expresiones inseguras

$$\{ t \mid \neg (t \in \text{loan}) \}$$

4.15.2 Cálculo relacional de dominios

account-number	branch-name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

account

branch-name	branch-city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

branch

customer-name	customer-street	customer-city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto

customer

customer-name	account-number
Hayes	A-102
Johnson	A-101

loan-number	branch-name	amount
L-11	RoundHill	900
L-14	Downtown	1500

customer-name	loan-number
Adams	L-16
Curry	L-93

Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

depositor

L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500
null	null	1900

loan

Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17
Johnson	null

borrower

- Encontrar branch-name, loan-number and amount para los préstamos superiores a \$1,200

$$\{ \langle l, b, a \rangle \mid \langle l, b, a \rangle \in \text{loan} \wedge a > 1200 \}$$

- Encontrar el loan-number de cada préstamo cuyo monto sea mayor a \$1,200

$$\{ \langle l \rangle \mid \exists b, a (\langle l, b, a \rangle \in \text{loan} \wedge a > 1200) \}$$

- Encontrar los nombres de los clientes que tienen un préstamo de la sucursal Perryridge

$$\{ \langle c, a \rangle \mid \exists l (\langle c, l \rangle \in \text{borrower} \wedge \exists b (\langle l, b, a \rangle \in \text{loan} \wedge b = \text{"Perryridge"})) \}$$

- Encontrar todos los clientes que tienen una cuenta en todas las sucursales localizadas en Brooklyn.

$$\{ \langle c \rangle \mid \exists n (\langle c, n \rangle \in \text{customer}) \wedge \forall x, y, z (\langle x, y, z \rangle \in \text{branch} \wedge y = \text{"Brooklyn"} \rightarrow \exists a, b (\langle a, x, b \rangle \in \text{account} \wedge \langle c, a \rangle \in \text{depositor})) \}$$

En este caso nuevamente aparece el "para todo" y el símbolo de implicación $P \rightarrow Q$, indicando que si P es cierto Q también debe serlo.