

Bases de Dados

T10 - Álgebra Relacional Parte I

Prof. Daniel Faria

Prof. Flávio Martins

Sumário

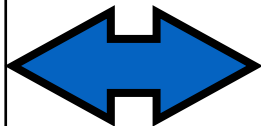
- Recapitulação Breve
- Álgebra Relacional
- Exemplos

Recapitulação Breve

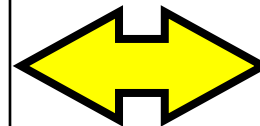
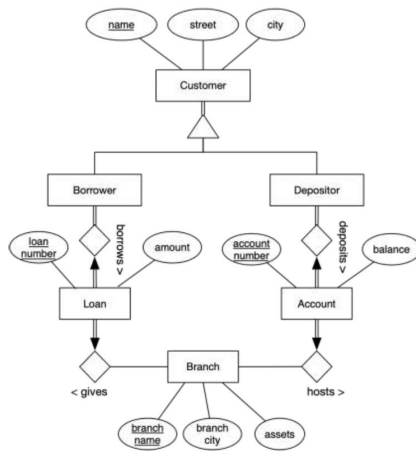
Concepção de Bases de Dados

Especificação de Requisitos

- requisito funcional 1:
- requisito funcional 2:
- ...
- restrição de integridade 1:
- restrição de integridade 2:
- ...



Modelo Conceptual



Modelo Relacional

R1 (x, y)
R2 (x, z)

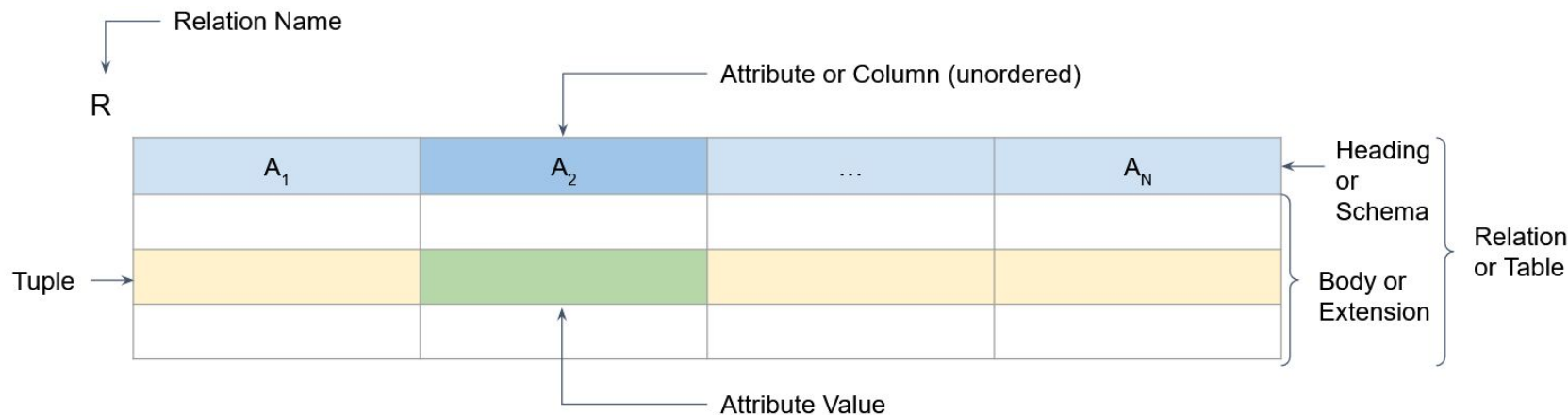
Esquema Relacional (SQL)

branch			account			depositor	
branch_name	branch_city	assets	account_number	branch_name	balance	customer_name	account_number
Brighton	Brooklyn	710000	A-101	Downtown	500	Hayes	A-101
Downtown	Brooklyn	900000	A-102	Perryridge	400	Johnson	A-201
Mannus	Horseneck	400000	A-201	Brighton	900	Jones	A-217
North Town	Rye	570000	A-215	Mannus	700	Lindsay	A-222
Perryridge	Horseneck	1700000	A-217	Brighton	750	Smith	A-215
Normal	Horseneck	300000	A-222	Redwood	798	Turner	A-305
Redwood	Palo Alto	2100000	A-305	Redwood	550		
Round Hill	Horseneck	800000					

loan			borrower			customer		
loan_number	branch_name	amount	customer_name	loan_number		customer_name	customer_city	customer_zip
L-11	Round Hill	900	Adams	L-16		Adams	Spring	Pittsfield
L-14	Downtown	1500	Brooks	L-15		Brooks	Senator	Brooklyn
L-15	Perryridge	1500	Curry	L-16		Curry	North	Rye
L-16	Perryridge	1200	Hayes	L-17		Glenn	Sand Hill	Woodside
L-17	Downtown	1000	Jackson	L-11		Gwen	Walnut	Stanford
L-23	Redwood	2000	Jones	L-12		Hayes	Main	Harrison
L-25	Mannus	500	Smith	L-13		Johnson	Palo Alto	Harrison
			Williams	L-17		Jones	Park	Pittsfield
						Lindsay	North	Rye
						Turner	Putnam	Stanford
						Williams	Nassau	Princeton

Relação

- Conjunto de tuplos n-ários que obedecem a uma especificação de nome e domínio de dados definida num cabeçalho
 - Representada normalmente como uma tabela



Conversão E-A–Relacional

- Entidades e associações E-A representam também conjuntos de tuplos de dados
- Conversão entre os dois modelos de dados visa:
 - Preservar todas as relações semânticas dos dados no modelo E-A
 - Minimizando a redundância de dados no modelo relacional

Conversão E-A–Relacional

Modelo E-A	Modelo Relacional
Entidade	Relação (mesmos atributos e mesma chave)
Associação 0-N–*	Relação (chaves das entidades associadas mais atributos da associação)
Associação 1-N–*	Relação + restrição de integridade para participação obrigatória
Associação 1–*	Chave da segunda entidade e atributos da associação incorporados na primeira entidade
Associação 0-1–*	Como 0-N mas chave é a primeira entidade ou como 1 (mas com NULLs)
Entidade Fraca	Relação com chave composta por chave parcial + chave da entidade forte
Generalização / Especialização	Relação para superclasse e subclasses (sempre possível) ou só subclasses (apenas se total) ou só superclasse (sempre possível mas com NULLs)
Agregação	Interior = associação; exterior = associação c/ agregação como entidade

Álgebra Relacional

Modelo Relacional (sem NULLs)

- Lógica de predicados de dois valores (verdadeiro ou falso)
- Selecção de dados baseada na teoria dos conjuntos usando:
 - Cálculo relacional: linguagem lógica declarativa com variáveis e quantificação; a base da optimização de queries
 - Álgebra relacional: linguagem procedimental imperativa sem variáveis; fundação teórica para o SQL
- Teorema de Codd: uma query pode ser formulada em cálculo relacional (independente do domínio) sse pode ser formulada em álgebra relacional

Motivação

- A Álgebra Relacional permite-nos pensar conceptualmente na interrogação a bases de dados relacionais
- Ajuda-nos a estruturar queries SQL

Álgebra Relacional

- Linguagem procedimental que consiste em operações algébricas sobre relações, cujo resultado é também uma relação
 - Inclui operações unárias (argumento é uma só relação) e binárias (argumento são duas relações)
- Uma vez que o resultado de cada operação é uma relação, operações de álgebra relacional podem ser combinadas em expressões

Álgebra Relacional

- Operadores básicos

- Seleção: σ
- Projeção: Π
- Produto Cartesiano: \times
- Renomear: ρ
- Diferença: $-$
- União: \cup

Sobre estes operadores podem-se definir outros, e.g.:

- Join: \bowtie
- Interseção: \cap
- Divisão: \div
- ...

Operadores de Álgebra Relacional

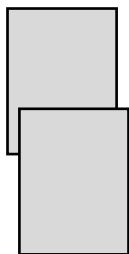
Select: σ

A	B	C	D

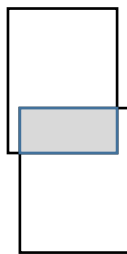
Project: Π

A	B	C	D

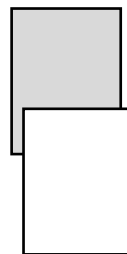
Union: \cup



Intersection: \cap



Difference: $-$



Cartesian Product: \times

a1	b1	b1	c1
a2	b2	b3	c2
a3	b3		



a1	b1	b1	c1
a1	b1	b3	c2
a2	b2	b1	c1
a2	b2	b3	c2
a3	b3	b1	c1
a3	b3	b3	c2

(Natural) Join: \bowtie

a1	b1	b1	c1
a2	b2	b3	c2
a3	b3		



a1	b1	c1
a3	b3	c2

Outer Join: $\ltimes, \ltimes, \ltimes$

a1	b1	b1	c1
a2	b2	b3	c2
a3	b3		



a1	b1	c1
a3	b3	c2
a2	b2	NULL

Division: \div

a1	b1	b1
a1	b2	b2
a2	b1	
a3	b2	



a1

Aggregation: G or γ

a1	
a2	
a3	
a4	



val

Seleção / Select

- Operação unária que seleciona tuplos de uma relação que satisfazem um determinado predicado
- Notação: $\sigma_p(r)$
 - em que p é o predicado de seleção
- E.g. seleciona todos os professores do departamento de Física

$$\sigma_{dept="Physics"}(professor)$$

ID	name	dept	salary
1	John	Chemistry	65000
2	Jack	Physics	95000
3	Jill	Physics	82000
4	Joan	Biology	82000



ID	name	dept	salary
2	Jack	Physics	95000
3	Jill	Physics	82000

Seleção / Select

- Um predicado de seleção deve ser booleano
- São permitidos os operadores de comparação: =, \neq , >, \geq , <, \leq
- Predicados podem ser compostos usando os operadores lógicos: \wedge (and), \vee (or), \neg (not)
- E.g. seleciona todos os professores do departamento de Física com salário > \$90,000

$$\sigma_{dept_name="Physics" \wedge salary > 90000} (professor)$$

ID	name	dept	salary
1	John	Chemistry	65000
2	Jack	Physics	95000
3	Jill	Physics	82000
4	Joan	Biology	82000



ID	name	dept	salary
2	Jack	Physics	95000

Seleção / Select

- Um predicado de seleção pode comparar um atributo com um literal (como nos exemplos anteriores) ou dois atributos
 - E.g. seleciona todos os departamentos cujo nome é igual ao nome do seu edifício
 - $\sigma_{dept_name=building}(department)$

Seleção / Select

Propriedades:

- A seleção é um operador comutativo:

$$\sigma_{p1}(\sigma_{p2}(r)) = \sigma_{p2}(\sigma_{p1}(r))$$

- Pode sempre reduzir-se uma série de seleções sobre uma relação a uma seleção única com todas as condições conjugadas com AND:

$$\sigma_{p1}(\sigma_{p2}(r)) = \sigma_{p1 \wedge p2}(r)$$

Projeção / Project

- Operação unária que devolve uma projeção da relação argumento num espaço dimensional inferior, i.e., apenas com os atributos listados
- Notação: $\Pi_{A1,A2,\dots,Ak}(r)$
 - onde cada Ai é um nome de atributo da relação r
- O resultado é definido como a relação obtida removendo da relação original todos os atributos não listados
 - Linhas duplicadas são removidas do resultado, uma vez que o resultado tem de ser uma relação

Projeção / Project

- E.g. obter apenas o salário (ou excluir tudo menos o salário) dos professores

$\Pi_{salary}(professor)$

- Notar que **não há garantia que os tuplos mantenham a ordem** (não há ordem nos tuplos de uma relação)
- Notar ainda a **exclusão do tuplo duplicado** “82000”

<i>ID</i>	<i>name</i>	<i>dept</i>	<i>salary</i>
1	John	Chemistry	65000
2	Jack	Physics	95000
3	Jill	Physics	82000
4	Joan	Biology	82000



<i>salary</i>
95000
82000
65000

Projeção / Project

Propriedades:

- A projeção **não** é um operador comutativo:

$$\Pi_{\langle list1 \rangle}(\Pi_{\langle list2 \rangle}(r)) \neq \Pi_{\langle list2 \rangle}(\Pi_{\langle list1 \rangle}(r))$$

- Porque a projeção $\Pi_{\langle x \rangle}(r)$ só está definida se $x \subseteq r$
- Pode sempre reduzir-se uma série de projeções “definidas” sobre uma relação a uma projeção única com os atributos da última projeção:

$$\Pi_{\langle list1 \rangle}(\Pi_{\langle list2 \rangle}(r)) = \Pi_{\langle list1 \rangle}(r) \text{ sse } list1 \subseteq list2 \subseteq r$$

Combinação de Operações

- Como referimos, o resultado de cada operação de álgebra relacional é uma relação
- Isto permite combinar várias operações em expressões, usando o resultado de uma operação como argumento de outra
- E.g. seleciona os nomes de todos os professores do departamento de Física

$$\Pi_{name} (\sigma_{dept_name="Physics"} (professor))$$

ID	name	dept	salary
1	John	Chemistry	65000
2	Jack	Physics	95000
3	Jill	Physics	82000
4	Joan	Biology	82000



name
Jack
Jill

Produto Cartesiano / Cross-Join

- Operação binária que combina todos os tuplos de uma relação com todos os tuplos de uma segunda relação
- Notação: $r \times s$
- E.g.: $professor \times teaches$

ID	name	dept	salary
1	John	Chemistry	65000
2	Jack	Physics	95000
3	Jill	Physics	82000
4	Joan	Biology	82000

4 tuplos, 4 atributos

ID	course
1	OCH-101
1	BCH-101
3	NPH-315

3 tuplos, 2 atributos



<i>professor.ID</i>	name	dept	salary	<i>teaches.ID</i>	course
1	John	Chemistry	65000	1	OCH-101
1	John	Chemistry	65000	1	BCH-101
1	John	Chemistry	65000	3	NPH-315
...
4	Joan	Biology	82000	1	OCH-101
4	Joan	Biology	82000	1	BCH-101
4	Joan	Biology	82000	3	NPH-315

$4 \times 3 = 12$ tuplos, $4 + 2 = 6$ atributos (renomeando iguais)

Produto Cartesiano / Cross-Join

Propriedades:

- O produto cartesiano é um operador comutativo

$$r \times s = s \times r$$

(relações não têm ordem de atributos nem de tuplos)

- O produto cartesiano é um operador associativo

$$r \times (s \times t) = r \times s \times t$$

Produto Cartesiano / Cross-Join

- O exemplo *professor* \times *teaches* contém todos os pares professor, ensina (relacionados ou não)
- Se queremos apenas os pares relacionados:

$\sigma_{\text{professor.id=teaches.id}} (\text{professor} \times \text{teaches})$

ID	name	dept	salary
1	John	Chemistry	65000
2	Jack	Physics	95000
3	Jill	Physics	82000
4	Joan	Biology	82000

ID	course
1	OCH-101
1	BCH-101
3	NPH-315



professor.ID	name	dept	salary	teaches.ID	course
1	John	Chemistry	65000	1	OCH-101
1	John	Chemistry	65000	1	BCH-101
3	Jill	Physics	82000	3	NPH-315

Join

- Operação binária que combina um produto cartesiano com uma seleção com um predicado sobre atributos
- Definição: $r \bowtie_{\theta} s = \sigma_{\theta}(r \times s)$
 - Em que θ é um predicado de atributos de $r \cup s$
- Portanto, o exemplo anterior pode ser reescrito como:
 - $\textit{professor} \bowtie_{\textit{professor.id=teaches.id}} \textit{teaches}$

Join

Propriedades:

- Join é um operador comutativo e associativo
- Há ainda “associatividade mútua” entre seleção e join / produto cartesiano quando o predicado de seleção apenas diz respeito a uma relação
- E.g. listar os cursos ensinados por professores do departamento de física

$\sigma_{dept_name="Physics"}(professor \bowtie_{professor.id=teaches.id} teaches)$

$\sigma_{dept_name="Physics"}(professor) \bowtie_{professor.id=teaches.id} teaches$

As duas queries não são idênticas são equivalentes

Natural Join

- Corresponde a um Join, em que:
 - θ é um predicado de igualdade entre todos os atributos com o mesmo nome nas duas relações
 - Atributos duplicados são removidos do resultado (projeção)
- Definição: $r \bowtie s = \Pi_{r \cup s} (\sigma_{r.A1=s.A1 \wedge \dots \wedge r.An=s.An} (r \times s))$
 - em que $\{A1, \dots, An\} = r \cap s$
- E.g. para duas relações $r(a,b,c,d)$ e $s(a,b,e,f,g)$

$$r \bowtie s = \Pi_{a,b,c,d,e,f,g} (\sigma_{r.a=s.a \wedge r.b=s.b} (r \times s))$$

Join vs. Natural Join

Join

professor ⋈_{*professor.id=teaches.id*} *teaches*

<i>professor.ID</i>	<i>name</i>	<i>dept</i>	<i>salary</i>	<i>teaches.ID</i>	<i>course</i>
1	John	Chemistry	65000	1	OCH-101
1	John	Chemistry	65000	1	BCH-101
3	Jill	Physics	82000	3	NPH-315

Natural Join

professor ⋈ *teaches*

<i>ID</i>	<i>name</i>	<i>dept</i>	<i>salary</i>	<i>course</i>
1	John	Chemistry	65000	OCH-101
1	John	Chemistry	65000	BCH-101
3	Jill	Physics	82000	NPH-315

Remoção de colunas duplicadas (já não é preciso desambiguar)

Renomear / Rename

- Operação unária que atribui um (novo) nome a uma relação e opcionalmente aos seus atributos
 - Tipicamente utilizada para dar nome à relação resultante de uma operação de álgebra relacional (que caso contrário é anónima)
- Notação: $\rho_x(r)$ ou $\rho_x r$
 - Atribui o nome x à relação r
- Notação: $\rho_{x(A1,A2,\dots,An)}(r)$
 - Atribui o nome x à relação r e renomeia os seus atributos (pela ordem que ocorrem) para $(A1,A2,\dots,An)$

Renomear / Rename

- Notação: $\rho_{x(3 \rightarrow A)}(r)$
 - Atribui o nome x à relação r e renomeia o terceiro atributo para A
- **NOTA:**
 - Quando há renomear envolvido, passamos a considerar a ordem dos atributos de uma relação (que teoricamente não existe)!
 - Quebra a comutatividade de operações que combinam relações (produto cartesiano e joins)
 - Torna a ordem dos atributos numa projeção relevante

Diferença / Set-Difference

- Operação binária que devolve os tuplos que estão numa primeira relação mas não numa segunda relação
- Notação: $r - s$
- A diferença só está definida para relações compatíveis, i.e., r e s têm de:
 - ter a mesma aridade
 - ter domínios de atributos compatíveis
- E.g. encontrar todos os cursos lecionados no primeiro semestre de 2022/2023 mas não no segundo

$$\Pi_{\text{course}}(\sigma_{\text{semester}="1" \wedge \text{year}="2022/2023"}(\text{edition})) - \Pi_{\text{course}}(\sigma_{\text{semester}="2" \wedge \text{year}="2022/2023"}(\text{edition}))$$

Diferença / Set-Difference

$\Pi_{\text{course}}(\sigma_{\text{semester}=1 \wedge \text{year}=\text{"2022/2023"}}(\text{edition})) -$
 $\Pi_{\text{course}}(\sigma_{\text{semester}=2 \wedge \text{year}=\text{"2022/2023"}}(\text{edition}))$

course	year	semester	coordinator
OCH-101	2021/2022	1	John
BCH-101	2021/2022	2	John
NPH-315	2022/2023	1	John
OCH-101	2022/2023	1	Joan
BCH-101	2022/2023	2	Joan
NPH-315	2022/2023	2	Joan



course
NPH-315
OCH-101



course
BCH-101
NPH-315



course
OCH-101

- **Propriedades:** a diferença não é comutativa nem associativa

União / Union

- Operação binária que devolve os tuplos que estão numa primeira relação ou numa segunda relação
- Notação: $r \cup s$
- Tal como a diferença, a união só está definida para relações compatíveis
- E.g. encontrar todos os cursos lecionados no primeiro ou segundo semestres de 2022/2023

$$\Pi_{\text{course}}(\sigma_{\text{semester}="1" \wedge \text{year}="2022/2023"}(\text{edition})) \cup \Pi_{\text{course}}(\sigma_{\text{semester}="2"}(\sigma_{\text{year}="2022/2023"}(\text{edition})))$$

União / Union

$\Pi_{\text{course}}(\sigma_{\text{semester}=1 \wedge \text{year}=\text{"2022/2023"}}(\text{edition})) \cup$
 $\Pi_{\text{course}}(\sigma_{\text{semester}=2 \wedge \text{year}=\text{"2022/2023"}}(\text{edition}))$

course	year	semester	coordinator
OCH-101	2021/2022	1	John
BCH-101	2021/2022	2	John
NPH-315	2022/2023	1	John
OCH-101	2022/2023	1	Joan
BCH-101	2022/2023	2	Joan
NPH-315	2022/2023	2	Joan



course
NPH-315
OCH-101

U

course
BCH-101
NPH-315

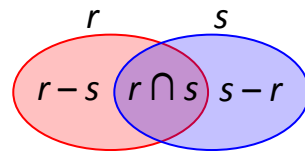


course
NPH-315
OCH-101
BCH-101

- **Propriedades:** a união é comutativa e associativa

Interseção / Set-Intersection

- Operação binária que devolve os tuplos que estão numa primeira relação e numa segunda relação
- Definição: $r \cap s = r \cup s - ((r - s) \cup (s - r))$
- Novamente, a interseção só está definida para relações compatíveis
- E.g. encontrar todos os cursos lecionados em ambos o primeiro e segundo semestres de 2022/2023



$$\prod_{\text{course}} (\sigma_{\text{semester}="1" \wedge \text{year}="2022/2023"}(\text{edition})) \cap \prod_{\text{course}} (\sigma_{\text{semester}="2" \wedge \text{year}="2022/2023"}(\text{edition}))$$

Interseção / Set-Intersection

$\Pi_{\text{course}}(\sigma_{\text{semester}=1 \wedge \text{year}=\text{"2022/2023"}}(\text{edition})) \cap$
 $\Pi_{\text{course}}(\sigma_{\text{semester}=2 \wedge \text{year}=\text{"2022/2023"}}(\text{edition}))$

course	year	semester	coordinator
OCH-101	2021/2022	1	John
BCH-101	2021/2022	2	John
NPH-315	2022/2023	1	John
OCH-101	2022/2023	1	Joan
BCH-101	2022/2023	2	Joan
NPH-315	2022/2023	2	Joan



course
NPH-315
OCH-101

\cap



course
BCH-101
NPH-315



course
NPH-315

- **Propriedades:** a interseção é comutativa e associativa

Exemplos

Base de Dados “Reservas de Barcos”

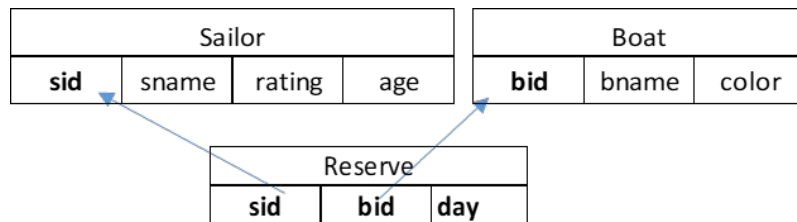
Sailors(sid, sname, rating, age)

Boats(bid, bname, color)

Reserves(sid, bid, day)

sid: FK(Sailors)

bid: FK(Boats)



Exemplo 1

- Quais os nomes dos marinheiros que reservaram o barco 33?

$$\Pi_{sname}(\sigma_{bid=33}(reserve \bowtie_{r.sid=s.sid} sailor))$$

Sailor			
sid	sname	rating	age
101	Maria	1	35
120	Zé	2	30
134	João	3	60

Reserve		
sid	bid	day
101	33	01/01/2020
120	30	11/01/2019
134	33	07/03/2019
101	20	11/11/2019

Reserve X Sailor						
r.sid	bid	day	s.sid	sname	rating	age
101	33	01/01/2020	101	Maria	1	35
101	33	01/01/2020	120	Zé	2	30
101	33	01/01/2020	134	João	3	60
120	30	11/01/2019	101	Maria	1	35
120	30	11/01/2019	120	Zé	2	30
120	30	11/01/2019	134	João	3	60
134	33	07/03/2019	101	Maria	1	35
134	33	07/03/2019	120	Zé	2	30
134	33	07/03/2019	134	João	3	60
101	20	11/11/2019	101	Maria	1	35
101	20	11/11/2019	120	Zé	2	30
101	20	11/11/2019	134	João	3	60

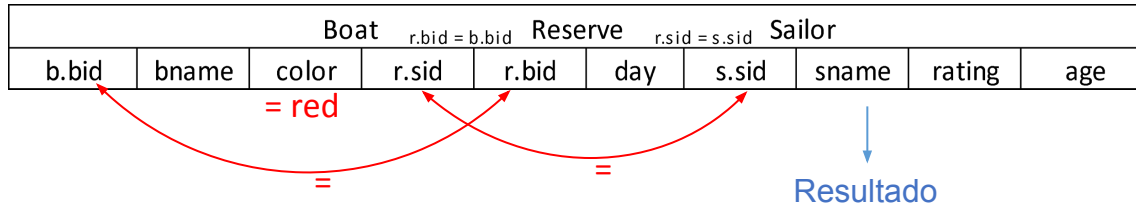
Reserve $r.sid = s.sid$ Sailor						
r.sid	bid	day	s.sid	sname	rating	age
101	33	01/01/2020	101	Maria	1	35
120	30	11/01/2019	120	Zé	2	30
134	33	07/03/2019	134	João	3	60
101	20	11/11/2019	101	Maria	1	35

sname
Maria
João

Exemplo 2

- Quais os nomes dos marinheiros que reservaram um barco vermelho?

$$\Pi_{sname}(\sigma_{color="red"}(boat) \bowtie_{b.bid=r.bid} reserve \bowtie_{r.sid=s.sid} sailor)$$



Exemplos 3-5

- Quais os nomes dos marinheiros que reservaram pelo menos um barco ?

$$\Pi_{sname}(reserve \bowtie_{r.sid=s.sid} sailor)$$

- Quais os nomes dos marinheiros que reservaram um barco vermelho ou verde?

$$\Pi_{sname}(\sigma_{color="red" \vee color="green"}(boat) \bowtie_{b.bid=r.bid} reserve \bowtie_{r.sid=s.sid} sailor)$$

- Quais os ids dos marinheiros com idade superior a 20 que não reservaram um barco vermelho?

$$\Pi_{sid}(\sigma_{age>20}(sailor)) - \Pi_{sid}((\sigma_{color="red"}(boat)) \bowtie_{b.bid=r.bid} reserve)$$

Exemplo 6

- Qual o marinheiro mais velho?

$$\underbrace{\prod_{sname}(sailor)}_{\text{Todos os marinheiros}} - \underbrace{\prod_{s1.sname}(\sigma_{s1.age < s2.age}(\rho_{s1} sailor \times \rho_{s2} sailor))}_{\text{Todos os marinheiros que são mais novos que outro marinheiro}}$$

S1 x S2							
s1.sid	s1.sname	s1.rating	s1.age	s2.sid	s2.sname	s2.rating	s2.age
101	Maria	1	35	101	Maria	1	35
101	Maria	1	35	120	Zé	2	30
101	Maria	1	35	134	João	3	60
120	Zé	2	30	101	Maria	1	35
120	Zé	2	30	120	Zé	2	30
120	Zé	2	30	134	João	3	60
134	João	3	60	101	Maria	1	35
134	João	3	60	120	Zé	2	30
134	João	3	60	134	João	3	60

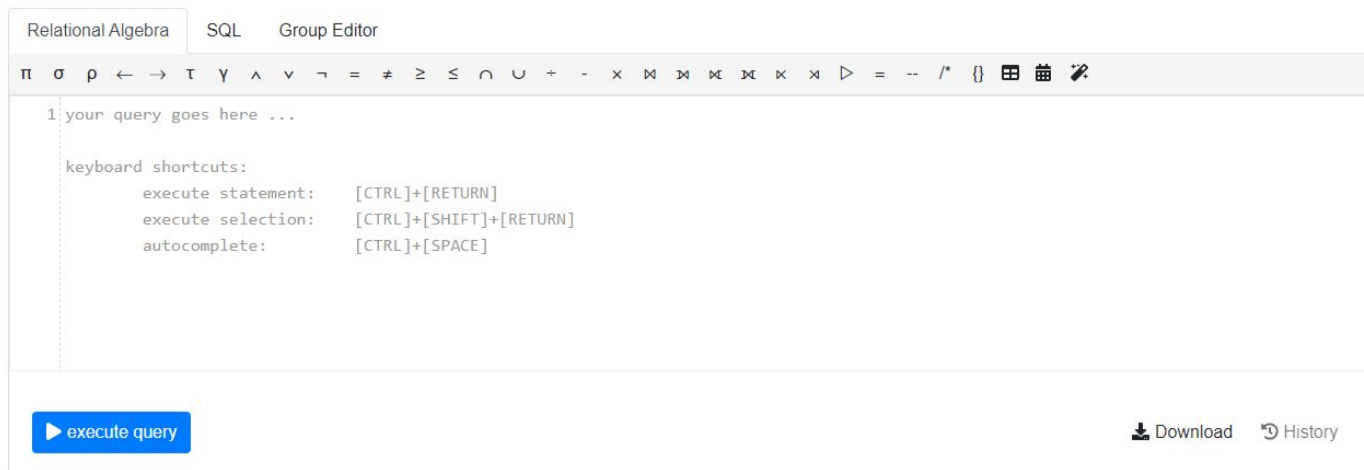
$\sigma_{S1.age < S2.age}(S1 \times S2)$							
s1.sid	s1.sname	s1.rating	s1.age	s2.sid	s2.sname	s2.rating	s2.age
101	Maria	1	35	134	João	3	60
120	Zé	2	30	101	Maria	1	35
120	Zé	2	30	134	João	3	60

s1.sname
Maria
Zé
João

s1.sname
Maria
Zé

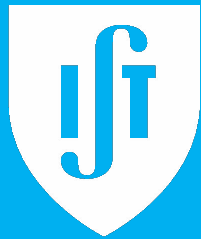
s1.sname
João

RelaX - relational algebra calculator



<http://dbis-uibk.github.io/relax/calc/local/uibk/local/6>

Use para praticar!



TÉCNICO LISBOA