

# Álgebra Relacional

Base de Dados - 2020/21  
 Carlos Costa

→ Precisamos de ferramentas para aceder à base de dados!,,  
 .lógica por traz dos acessos...

1

## Introdução

### Linguagem de Consulta/Interrogação de BD

- Álgebra Relacional → Teoria de Conjuntos!,,
  - Linguagem formal do Modelo Relacional
  - Um conjunto básico de operações
- Outras linguagem formais: *relational calculus*
- As linguagens formais oferecem uma base teórica para a linguagem de consulta utilizada na prática.
- Linguagem prática do Modelo Relacional
  - SQL

2

2

**Álgebra Relacional**  
→ Álgebra sobre conjuntos

Questões?

- Como deve ser uma linguagem de interrogação da BD?
- Que tipo de interrogações existem?
- Como é que são os resultados?
- Expressões de álgebra relacional (linguagem).
  - Sequência de operações de álgebra relacional.
  - Permitem formular pedidos básicos de recuperação de informação sobre uma ou mais relações.
- Formulação da interrogação:
  - conjunto de operadores que operam sobre as relações
  - devolvem uma nova relação
- Vamos estudar um conjunto de operações...

3

3

**Álgebra Relacional - Operações Básicas**

$\sigma$	• Seleção	
$\pi$	• Projeção	
$\cup$	• União	
$-$	• Diferença	
$\times$	• Produto Cartesiano	
$\rho$	• Renomeação	

4

- $\div$  • Divisão
- $\bowtie$  • Inner Join

(...)

2

## Seleção



"sigma"

- Notação:  $\sigma_{<\text{selection condition}>}(R)$

- Utilizada para selecionar um subconjunto de tuplos da relação ( $t \in R$ ) que satisfazem os critérios de seleção.
- “selection condition” é uma expressão booleana.

e.g.:  
Resultado  $\leftarrow \sigma_{Dno=5}(\text{Employee})$   
↑ set  
↓ set

$\text{Relation2} \leftarrow \sigma_{<\text{selection condition}>}(\text{Relation1})$

- O resultado é uma nova relação (Relation2) que tem um esquema relacional igual à original (Relation1).

5

5

## Seleção - Predicado

- Operadores de Comparação

- Permitem comparar dois atributos ou um atributo com um valor.
- Operandos: Nomes dos atributos e constantes.

- Operadores:  $=, \neq, \leq, \geq, <, >$

- Exemplos:

$\sigma_{Dno=4}(\text{EMPLOYEE})$

$\sigma_{\text{Salary}>30000}(\text{EMPLOYEE})$

- Condições Booleanas

- Utilização de AND, OR e NOT.

- Exemplo:

$\sigma_{(Dno=4 \text{ AND } \text{Salary}>25000) \text{ OR } (Dno=5 \text{ AND } \text{Salary}>30000)}(\text{EMPLOYEE})$

6

6

**Seleção - Exemplo**

**EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

**SQL query (próxima aula...)**

$\sigma_{(Dno=4 \text{ AND } Salary > 25000) \text{ OR } (Dno=5 \text{ AND } Salary > 30000)}(\text{EMPLOYEE})$

**SELECT \* FROM EMPLOYEE  
WHERE Dno=4 AND Salary>25000  
OR Dno=5 AND Salary>30000;**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5

7

7

**Projeção**

• Notação:  $\Pi_{<\text{attribute list}>}(\text{R})$

- $<\text{attribute list}> = A_1, A_2, \dots, A_k$  (A condição de set pode não se verificar... (eliminam linhas repetidas))
- $A_1 \dots A_k$  são nomes dos atributos da relação R

• O resultado é uma nova relação só com os k atributos selecionados.

Podemos não ter uma chave candidata na projeção ⇒ Podem haver linhas repetidas

• São removidas as linhas duplicadas do resultado.

- Condição de conjunto (set)

↳ Na lógica isto acontece! //

8

8

⇒ Mas na linguagem prática temos de fazer isto

☒ ☐ ☒

**Projeção - Exemplo**

**EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

$\Pi_{Lname, Fname, Salary} (EMPLOYEE)$

**SQL query:**

```
SELECT DISTINCT Lname, Fname, Salary
FROM EMPLOYEE;
```

**Result:**

Lname	Fname	Salary
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

9

Garante que só  
eliminados os  
repetidos

9

**Encadeamento de Operações**

- $\Pi_{Fname, Lname, Salary} (\sigma_{Dno=5}(EMPLOYEE))$

- Se quisermos renomear os atributos e a relação:

- ①  $TEMP \leftarrow \sigma_{Dno=5}(EMPLOYEE)$
- ②  $R(First\_name, Last\_name, Salary) \leftarrow \Pi_{Fname, Lname, Salary}(TEMP)$

*Operação de renomeração*

**TEMP**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

**R**

First_name	Last_name	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

10

10

## Renomeação

• Não só por estética!  deti  
• Para as operações pode ser bom...

- Notação:  $\rho_{R2(B1, B2, \dots, Bn)}(R1)$  ou  $\rho_{R2}(R1)$   
↑ "nó"
  - Nome + os atributos!!!
  - Mude o nome da tabela
- ou  $\rho_{(B1, B2, \dots, Bn)}(R1)$   
Mude os atributos!

- No primeiro caso o resultado é uma nova relação R2 com os atributos renomeados (B1, B2, ..., Bn).
- No segundo caso só renomeamos a relação.
- No terceiro só renomeamos os atributos.

**SQL query:**

```
SELECT E.Fname AS First_name, E.Lname AS Last_name, E.Salary AS Salary
FROM EMPLOYEE AS E
WHERE E.Dno=5;
```

**Seleção**

R1: EMPLOYEE  
R2: E  
Fname -> First\_name  
Lname -> Last\_Name  
...

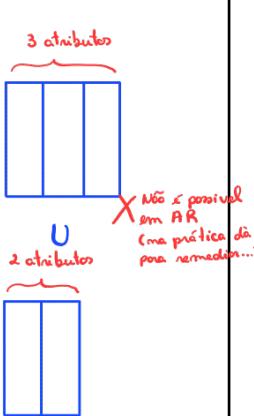
11

11

## União

• Notação:  $R \cup S = \{t : t \in R \vee t \in S\}$

- As tabelas têm de ser compatíveis
  - Mesmo número de atributos
  - Atributos com domínios compatíveis
- O resultado é uma relação que inclui todos os tuplos de R e de S
  - Os tuplos duplicados são eliminados



**U**

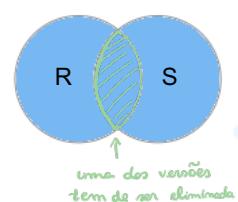
**INSTRUTOR**

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

**STUDENT**

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

**U**



uma das versões tem de ser eliminada

12

12

Mesmo número de atributos e  
têm de ser compatíveis

## Intersecção

 deti

- Notação:  $R \cap S = \{t : t \in R \wedge t \in S\}$
- As tabelas têm de ser compatíveis
  - Mesmo número de atributos
  - Atributos com domínios compatíveis  $\neq$
- O resultado é uma relação que inclui os tuplos que existem simultaneamente em R e S
  - Os tuplos duplicados são eliminados

STUDENT	
Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

$\cap$

INSTRUCTOR	
Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

$\rightarrow$

Fn	Ln
Susan	Yao
Ramesh	Shah

13

13

## Diferença

 deti

- Notação:  $R - S = \{t : t \in R \wedge t \notin S\}$
- As tabelas têm de ser compatíveis
  - Mesmo número de atributos
  - Atributos com domínios compatíveis  $\neq$
- O resultado é uma relação que inclui os tuplos de R que não existem em S

STUDENT	
Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

-

INSTRUCTOR	
Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

$\rightarrow$

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

14

14

**União, Intersecção e Diferença**

deti

- Em SQL existem os seguintes comandos
  - UNION (ALL), INTERSECT (ALL) e EXCEPT (ALL)

*Reserva ou não os duplicados*

Propriedades:

- União e Intersecção são operações comutativas:
  - $R \cup S = S \cup R$  e  $R \cap S = S \cap R$
- A diferença não é comutativa:
  - $R - S \neq S - R$  → A diferença não é comutativa!»
- União e Intersecção são operações associativas:
  - $R \cup (S \cup T) = (R \cup S) \cup T$  e  $(R \cap S) \cap T = R \cap (S \cap T)$

15

15

**Produto Cartesiano** *→ Computacionalmente despendiosa!!!*

- Notação:  $R \times S$
- Permite-nos combinar tuplos de relações diferentes.
  - O resultado é uma nova relação (Q) que combina cada elemento (tuplo) de uma relação (R) com um elemento (tuplo) da outra relação (S):
 
$$Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m) = R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$$
  - O número de tuplos de Q é  $n * m$ .
- UK: “CROSS JOIN”

16

16



**Produto Cartesiano - Exemplo**

Para "explorar" um servidor é bom!

**EMPNAME**

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

**DEPENDENT**

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

**EMPNAME X DEPENDENT**

EMPNAME X DEPENDENT					
Alicia	Zelaya	999887777	333445555	Alice	F
Alicia	Zelaya	999887777	333445555	Theodore	M
Alicia	Zelaya	999887777	333445555	Joy	F
Alicia	Zelaya	999887777	333445555	Abner	M
Alicia	Zelaya	999887777	123456789	Michael	M
Alicia	Zelaya	999887777	123456789	Alice	F
Alicia	Zelaya	999887777	123456789	Elizabeth	F
Jennifer	Wallace	987654321	333445555	Alice	F
Jennifer	Wallace	987654321	333445555	Theodore	M
Joyce	English	453453453	333445555	Joy	F
Joyce	English	453453453	333445555	Abner	M
Joyce	English	453453453	333445555	Michael	M
Joyce	English	453453453	333445555	Alice	F
Joyce	English	453453453	123456789	Elizabeth	F

17

17

**Junção  $\theta$  (THETA JOIN)**

*lembre que:*  
 → A união normal é na vertical!  
 → Aqui estamos na horizontal!

*O Dino aí chega estranhamente!!! Estamos de volta! Com 2 tabelas,*

*Employee  $\bowtie$  Department*  
*Dino = Dmnumber*

*e.g.: T  $\leftarrow$  (Employee  $\bowtie$  Department) (Employee x Department)*

*Inserimos ficar só com estes...*

**Notação:  $R \bowtie_C S$**

- Pode ser visto como o resultado das seguintes operações:
  - $R3 \leftarrow R1 \times R2$  (produto cartesiano)
  - $\sigma_C(R3)$  (seleção com condição c)

*Todos os casos possíveis em que ' $C$ '.*

- $C$  é *<join condition>* que pode tomar a seguinte forma:
  - <condition> AND <condition> AND ... AND <condition>* Qualquer condição!
- Em cada *<condition>* podemos aplicar operadores de comparação:  
 $=, <, \leq, >, \geq, \neq$

18

18

**Junção θ (THETA JOIN) - Exemplo**

deti

- Pretendemos saber os nomes dos funcionários gestores de departamentos

EMPLOYEE									
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1985-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1985-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Balaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1982-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Para obter o nome dos gestores temos de combinar cada tuplo do departamento (Department) com um tuplo dos funcionários (Employee) cujo Ssn é igual ao Mgr\_ssn.

**DEPT\_MGR ← DEPARTMENT ⋈<sub>Mgr\_ssn=Ssn</sub> EMPLOYEE** → Vou buscar o nome dos gestores!

DEPT_MGR								
Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

Depois só temos de utilizar projeção para obter os atributos desejados:

**RESULT ← π<sub>Dname, Lname, Fname</sub>(DEPT\_MGR)**

19

**Junção - Variações da Junção θ**

deti

- Equi-Junção (EquiJoin)**
  - É utilizado o operador = na condição de junção.
  - Exemplo anterior: DEPARTMENT ⋈<sub>Mgr\_ssn=Ssn</sub> EMPLOYEE.
  - Vamos ter sempre duas colunas repetidas.
- Junção Natural (Natural Join): R ⋈ S**
  - Condição implícita: igualdade dos atributos com o mesmo nome.
  - Os atributos repetidos são removidos.
  - Nota: Muitas vezes opta-se por renomear colunas de modo a facilitar junções naturais.

**R**

X	Y
a	c
b	d

**S**

Y	Z
d	g
e	h

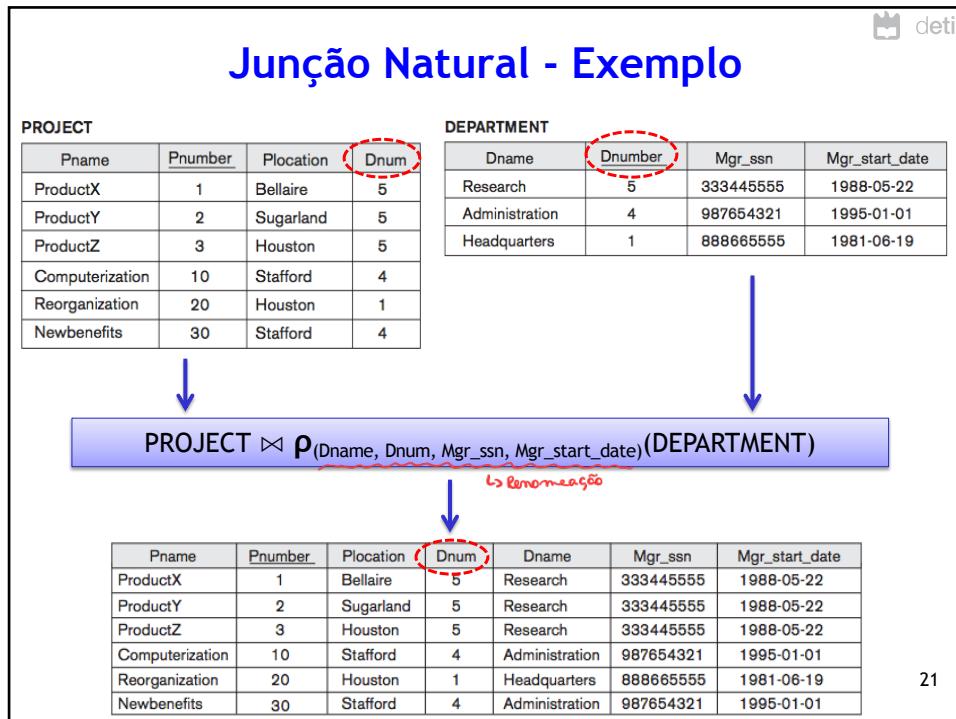
→

X	Y	Z
b	d	g

*Não tem condição!  
(ele assume que é nos atributos com o mesmo nome)*

*Não existem colunas repetidas!//*

20



21

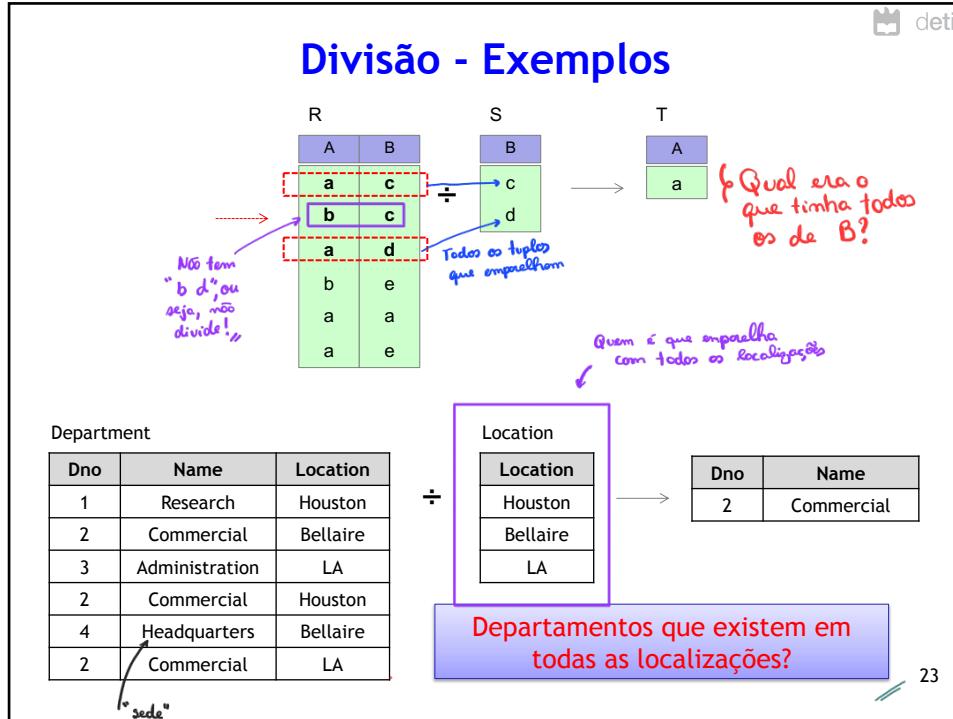
**Divisão**

• Não existe sobre conjuntos!

- Notação:  $R \div S$
- Dadas as relações  $R(A_1, \dots, A_r, B_1, \dots, B_k)$  e  $S(B_1, \dots, B_k)$
- O resultado incluirá todos os tuplos de  $R(A_1, \dots, A_r)$  que tenham correspondência com todos os tuplos de  $S$  em  $R(B_1, \dots, B_k)$ .
- $R_1$  e  $R_2$  são projeções de  $R$
- número de atributos de  $R >$  número de atributos de  $S$ .
- Em SQL não existe um operador que implemente a divisão. Temos de recorrer a operadores básicos:
- $R \div S = \pi_{R-S}(R) - \pi_{R-S}((\pi_{R-S}(R) \times S) - R)$
- onde  $\pi_{R-S} \rightarrow \pi_{(A_1, \dots, A_r)}$

22

22



23

**Operações Álgebra Relacional - Resumo**

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation $R$ .	$\sigma_{<\text{selection condition}>}(R)$
PROJECT	Produces a new relation with only some of the attributes of $R$ , and removes duplicate tuples.	$\pi_{<\text{attribute list}>}(R)$
THETA JOIN	Produces all combinations of tuples from $R_1$ and $R_2$ that satisfy the join condition.	$R_1 \bowtie_{<\text{join condition}>} R_2$
EQUIJOIN	Produces all the combinations of tuples from $R_1$ and $R_2$ that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{<\text{join condition}>} R_2$ , OR $R_1 \bowtie_{(<\text{join attributes 1}>), (<\text{join attributes 2}>)} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of $R_2$ are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{<\text{join conditions}>} R_2$ , OR $R_1 *_{(<\text{join attributes 1}>), (<\text{join attributes 2}>)} R_2$ , OR $R_1 *_{R_2} R_2$
UNION	Produces a relation that includes all the tuples in $R_1$ or $R_2$ or both $R_1$ and $R_2$ ; $R_1$ and $R_2$ must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both $R_1$ and $R_2$ ; $R_1$ and $R_2$ must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in $R_1$ that are not in $R_2$ ; $R_1$ and $R_2$ must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of $R_1$ and $R_2$ , and includes tuples all possible combinations of tuples from $R_1$ and $R_2$ .	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in $R_1$ in combination with every tuple from $R_2(Y)$ , where $Z = X \cup Y$ .	$R_1(Z) \div R_2(Y)$

24

24

## Álgebra Relacional - Operações Estendidas

- Semi-Join (Semi Junção)
  - Left Semi Join
  - Right Semi Join
- Outer Join (Junção Externa)
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join
- Agregação
  - Funções de Agregação  Começa a complicar! „

25

25

## Semi Join

- Só interessado nos dados do lado do R!!!*
- Left Semi Join:  $R \ltimes S = \Pi_R(R \bowtie S)$
- Projeção dos atributos de R na junção natural de R com S

R	S	
x a b	y c d	
	d e h	

→

x	y
b	d

- Right Semi Join:  $R \rtimes S = \Pi_S(R \bowtie S)$

Projeção dos atributos de S na junção natural de R com S

R	S	
x a b	y c d	
	e h	

→

y	z
d	g

26

26



Ssn	Fname	Lname	...	Dno
1234	Carlos	Costa	...	5
3452	Ana	Maria	...	1
...	...	...	...	...
5215	Rui	Silva	-	NULL

Department	Dname	...	...
1	Comercial	...	...
2	R&D	...	...
3	Marketing	...	...
5	Gestão	...	-

Employee  $\bowtie$  Department  
Dno=Dno

4/18/21

Ssn	Fname	Lname	...	Dno	Dname	...	...
1234	Carlos	Costa	...	5	5	Gestão	...
3452	Ana	Maria	...	1	1	Comercial	...
...	...	...	...	...	...	...	...
5215	Rui	Silva	-	NULL	NULL	NULL	NULL

Tuplo que não emparelha!!!

Fiz com a NULL

## Inner Join vs Outer Join

### Inner Join

- As operações de junção anteriores combinam dados de duas tabelas para que estes possam ser apresentados na forma de uma única tabela.
- Os tuplos que não estão relacionados (*matching*) são descartados.
  - Incluindo os tuplos com valores Null nos atributos de junção.

### Em Full Outer Join:

- Inclui os NULL's do lado esquerdo também!
- E o Left + Right Outer Join

→ Dá-me os departamentos que não têm funcionários!  
→ Right Outer Join!

### Outer Join

- Incluímos no resultado todos os tuplos de uma (ou de ambas) das relações componentes.
- Os atributos que não fazem *matching* são preenchidos com *Null*.

27

27

## Outer Join

### • Left Outer Join: R $\bowtie_{A2=B1}^L$ S

↑ obrigatório

R		S	
A1	A2	B1	B2
a	c	d	g
b	d	e	h

$\bowtie_{A2=B1}$

### • Right Outer Join: R $\bowtie_{A2=B1}^R$ S

R		S	
A1	A2	B1	B2
a	c	d	g
b	d	e	h

$\bowtie_{A2=B1}^R$

### • Full Outer Join: R $\bowtie_{A2=B1}^F$ S

?

R		S	
A1	A2	B1	B2
a	c	d	g
b	d	e	h

$\bowtie_{A2=B1}^F$

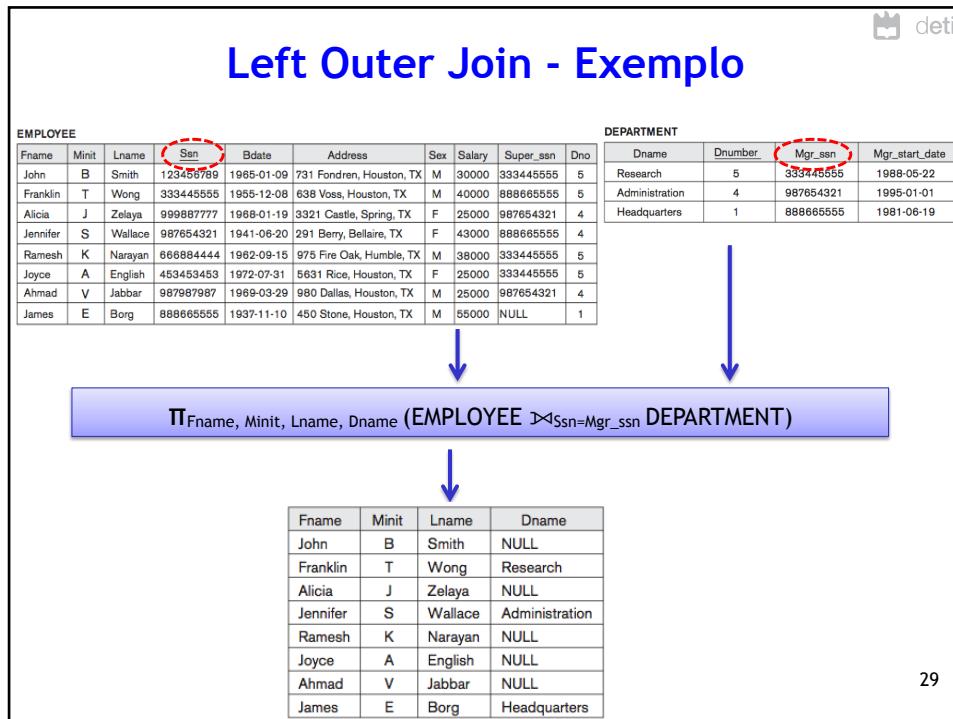
→ Dá-me os departamentos que não têm funcionários!

TT\_dome (Employee DE Department  
Dno=Dno)  
→ os mesmos!!!

28

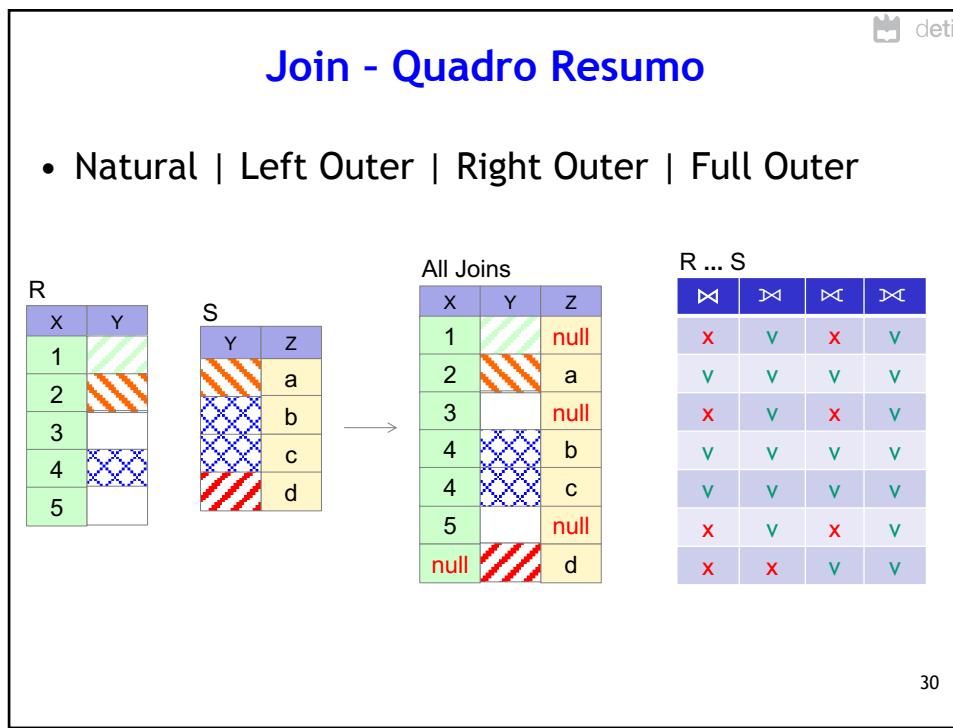
28

14



29

29



30

30

## Agregação

- Operação de Agregação
 

$\langle \text{grouping attributes} \rangle \setminus \langle \text{function list} \rangle (R)$
- Operações sobre vários tuplos da relação
- Lista de Funções de Agregação:
  - **avg**: média dos valores
  - **min**: mínimo dos valores
  - **max**: máximo dos valores
  - **sum**: soma dos valores
  - **count**: número dos valores

31

"Business Intelligence"

→ Perceber como estão os vendas...  
→ Como evoluem os salários ...

Quero:

Dno	Média
Gestão	1000
Deti	5000

$\setminus$   $\Pi_{Dno, Avg\_Salary = avg(Salary)} (Employee)$

$\Sigma$  - Script F symbol

$\Pi_{Dno, Avg\_Salary = avg(Salary)} (Employee)$

31

## Funções de Agregação

- Também podem ser usadas em projeções
  - criar atributos agregados
  - os atributos não agregados são agrupados de forma a não haver valores repetidos.
- Exemplos:
 

$\Pi_{A1, A2, M = avg(A3)} (R)$

EMPLOYEE											
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno		
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5		
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5		
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4		
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4		
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5		
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5		
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4		
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1		

$\Pi_{Dno, Avg\_Salary = avg(Salary)} (EMPLOYEE)$

Dno	Avg_Salary
1	55000
4	31000
5	33250

32

32

deti

## Agregação (*Grouping*) - Exemplos

**EMPLOYEE**

Name	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

$\exists \text{ count}(\text{Ssn}), \text{ avg}(\text{Salary})(\text{EMPLOYEE})$

Count_ssn	Average_salary
8	35125

$(Dno) \exists \text{ count}(\text{Ssn}), \text{ avg}(\text{Salary})(\text{EMPLOYEE})$

*"Pedia pra mais do que um!!"*

$\rho_{R(Dno, No\_of\_employees, Average\_sal)}(Dno \exists \text{ count}(\text{Ssn}), \text{ avg}(\text{Salary})(\text{EMPLOYEE}))$

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

33

33

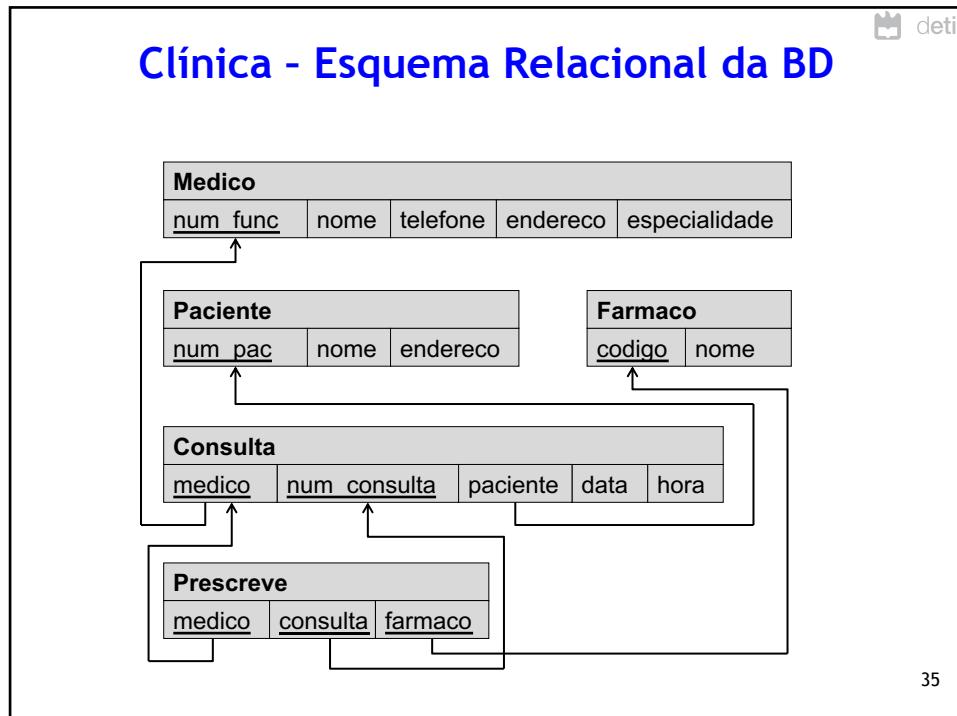
deti

## Álgebra Relacional - Queries Caso de Estudo

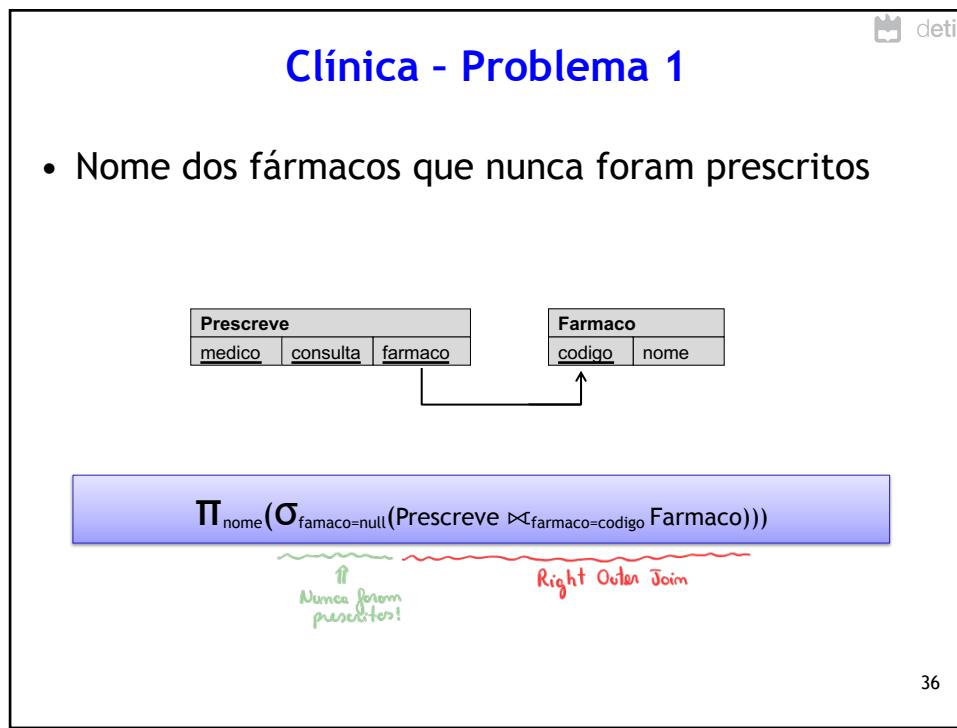
### Clínica Médica

34

34



35



36

**deti**

## Clínica - Problema 2

- O número de fármacos prescritos em cada consulta

Prescreve			Farmaco	
medico	consulta	farmaco	codigo	nome

↑

$\Pi_{medico, consulta, num\_farm=count(farmaco)} (\text{Prescreve})$

Ou

$\Pi_{medico, consulta} \Sigma_{farmaco} count(farmaco) (\text{Prescreve})$

37

medico	Consulta	Farmaco
A	1	521
A	1	329
A	2	321
A	3	541
A	3	425
-	-	-

medico, consulta  $\Sigma_{farmaco} count(farmaco)$  (Prescreve)

37 (Prescreve)

→

medico	Consulta	N Farmacos
A	1	2
A	2	1
A	3	2

**deti**

## Clínica - Problema 3

- Para cada médico, a quantidade média de fármacos receitados por consulta

Prescreve			Farmaco	
medico	consulta	farmaco	codigo	nome

↑

$\text{temp} \leftarrow \Pi_{medico, consulta, num\_farm=count(farmaco)} (\text{Prescreve})$

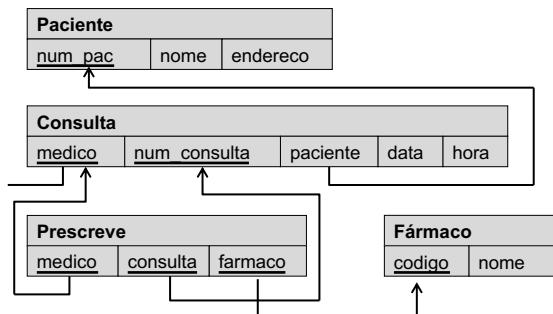
$\Pi_{medico, avg\_farmaco=avg(num\_farm)} (\text{temp})$

38

38

## Clínica - Problema 4

- O nome de todos os fármacos prescritos, incluindo a quantidade, para o paciente número 35312161



$\text{temp} \leftarrow \Pi_{\text{medico}, \text{num\_consulta}} (\sigma_{\text{paciente}=35312161} (\text{Consulta}))$

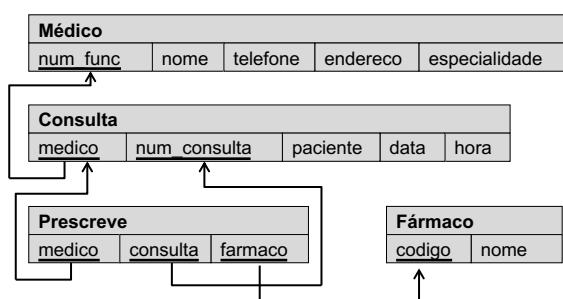
$\text{temp2} \leftarrow \Pi_{\text{farmaco}, \text{quantidade}=\text{count}(\text{farmaco})} (\text{temp} \bowtie_{\text{medico}=\text{medico} \text{ AND } \text{num\_consulta}=\text{consulta}} \text{Prescreve})$

$\Pi_{\text{nome}, \text{quantidade}} (\text{temp2} \bowtie_{\text{farmaco}=\text{codigo}} \text{Farmaco})$

39

## Clínica - Problema 5

- O nome dos fármacos que já foram prescritos por todos os médicos da clínica

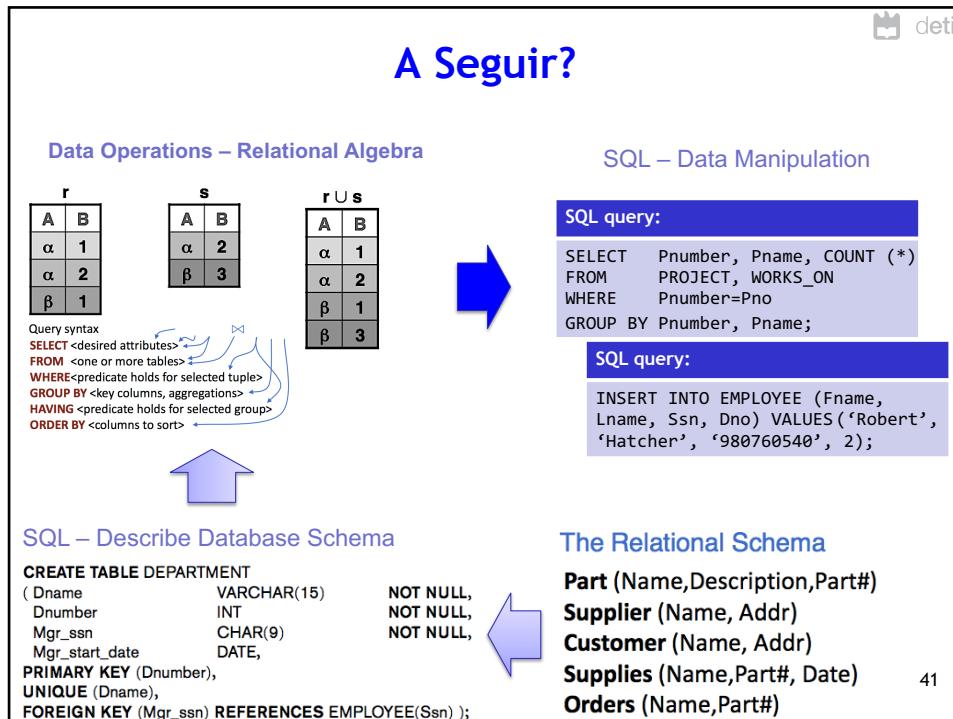


$\text{temp} \leftarrow (\Pi_{\text{farmaco}, \text{medico}} (\text{Prescreve})) \div (\rho_{\text{medico}} (\Pi_{\text{num_func}} (\text{Médico})))$

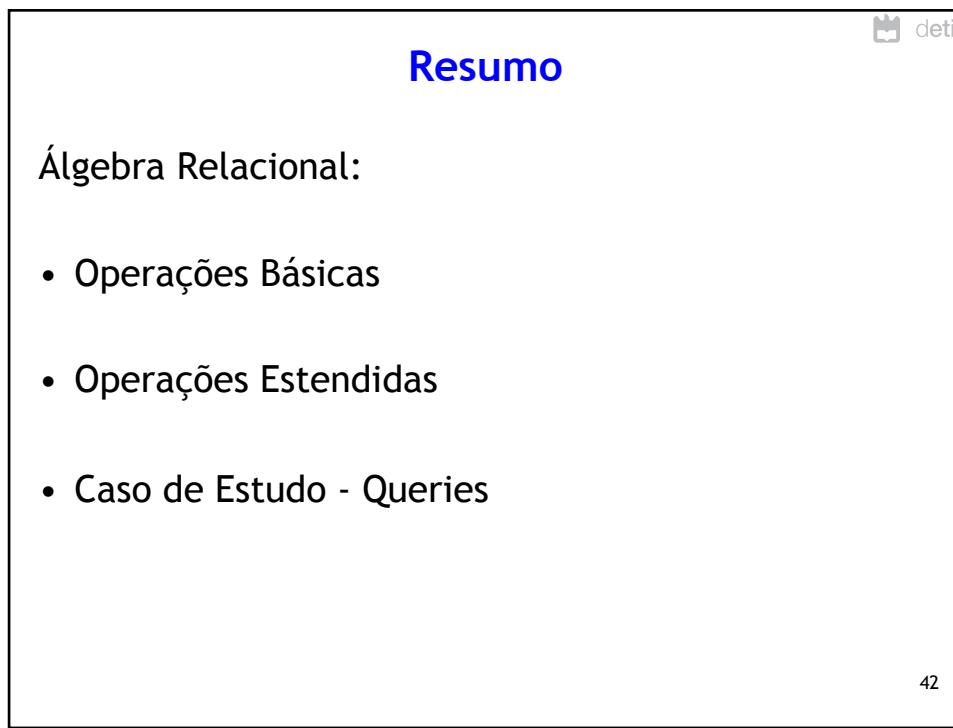
$\Pi_{\text{nome}} (\rho_{\text{codigo}, \text{medico}} (\text{temp}) \bowtie \text{Farmaco})$

40

40



41



42

42