

# Algebra Relacional

Mestrado BioInformática, 1º ano

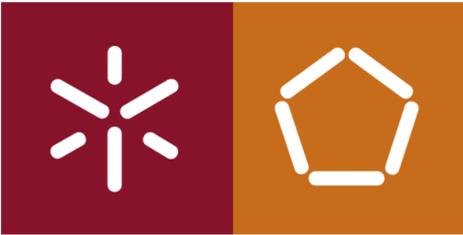
Unidade Curricular de ... Bases de Dados

Aula 4

António Abelha

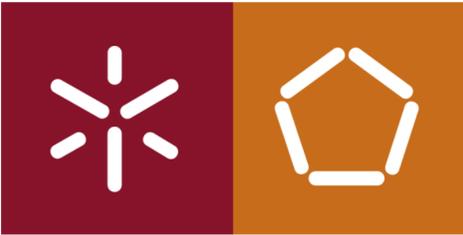
Departamento de Informática

Escola de Engenharia



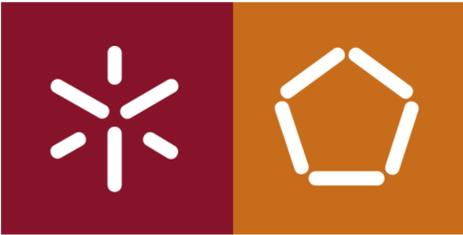
# Álgebra Relacional

- A álgebra relacional é uma linguagem teórica com operações sobre uma ou mais relações para definir uma nova relação, sem alterar as relações iniciais.
- Ambos os operandos e os resultados são relações, assim a saída de uma operação pode tornar-se a entrada para outra operação. Isso permite que expressões possam ser aninhadas.
- Esta propriedade é chamada de fecho, as relações estão fechadas sob a álgebra, assim como as operações numéricas estão fechados sob as operações aritméticas.



# Álgebra Relacional

- Há muitas variações das operações que estão incluídos na álgebra relacional. Codd originalmente propôs oito operações.
- As cinco operações fundamentais em álgebra relacional, **Seleção**, **Projeção**, **Produto cartesiano**, **União**, e **Diferença**. Além destas temos também a **Junção**, operações de **Interseção** e **Divisão** que podem ser expressas em termos das cinco operações básicas.



# Álgebra Relacional

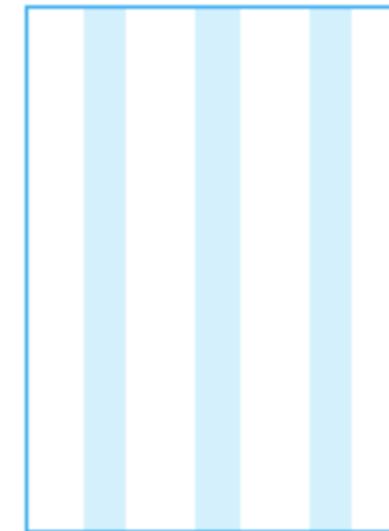
- Há muitas variações das operações que estão incluídos na álgebra relacional. Codd originalmente propôs oito operações.
- As cinco operações fundamentais em álgebra relacional, **Seleção**, **Projeção**, **Produto cartesiano**, **União**, e **Diferença**. Além destas temos também a **Junção**, operações de **Interseção** e **Divisão** que podem ser expressas em termos das cinco operações básicas.

# Algebra Relacional

- **Operações unárias**



(a) Selection



(b) Projection

$$\sigma_{\text{predicate}}(\mathbf{R})$$

- A operação de selecção funciona numa única relação  $R$  e define uma relação que contém apenas as tuplos de  $R$  que satisfazem a condição (predicate).

$$\Pi_{a_1, \dots, a_n}(\mathbf{R})$$

- A operação de projecção funciona numa única relação  $R$  e define um relação que contém um subconjunto vertical de  $R$ , extrai os valores dos atributos especificados eliminando duplicados.

# Algebra Relacional

## Operações unárias (exemplos)

Listar os funcionários com um salário maior do que 10.000.

$$\sigma_{\text{salary} > 10000} (\text{Staff})$$

A relação de entrada é Staff e o predicado é salary > 10000. A operação Seleção define uma relação contendo apenas os tuplos **Staff** com um salário superior a 10.000. Para predicados complexos podemos utilizar os operadores lógicos  $\wedge$  (AND),  $\vee$  (OR) e  $\sim$  (NOT).

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

# Algebra Relacional

## Operações unárias (exemplos)

Listar os salários para todos os funcionários, mostrando apenas o *staffNo*, *fName*, *IName*, e *salary*.

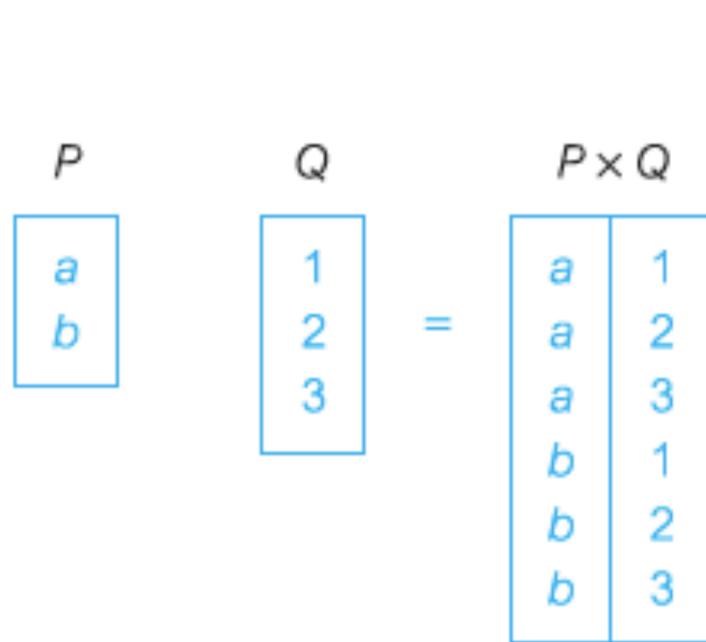
$\Pi_{\text{staffNo, fName, IName, salário}}(\text{Staff})$

Neste exemplo, a operação de projecção define uma relação que contém apenas os seguintes atributos da relação *Staff*: *staffNo*, *fName*, *IName*, e *salary*, na ordem especificada.

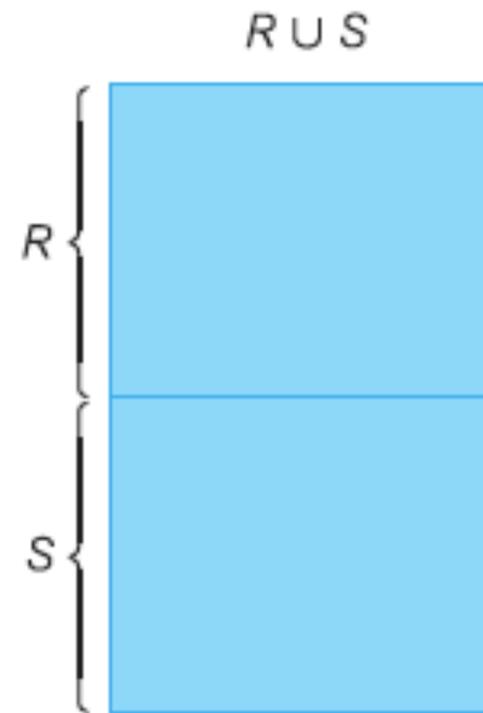
staffNo	fName	IName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000

# Algebra Relacional

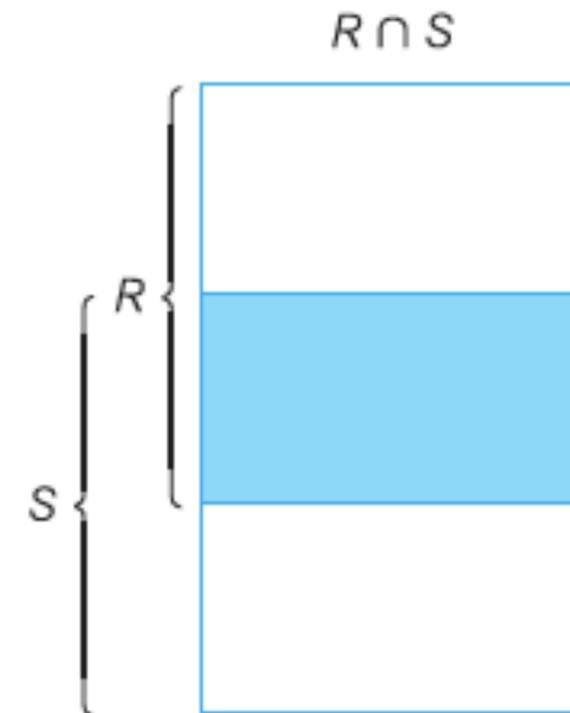
## Operações Conjuntos



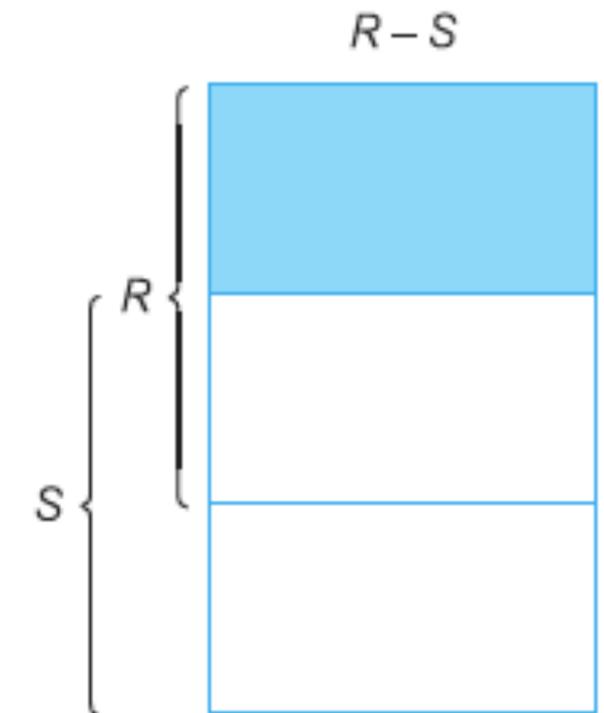
(c) Cartesian product



(d) Union



(e) Intersection



(f) Set difference

# Algebra Relacional

## Operações Conjuntos

*T*

A	B
a	1
b	2

*U*

B	C
1	x
1	y
3	z

$T \bowtie U$

A	B	C
a	1	x
a	1	y

$T \triangleright_B U$

A	B
a	1

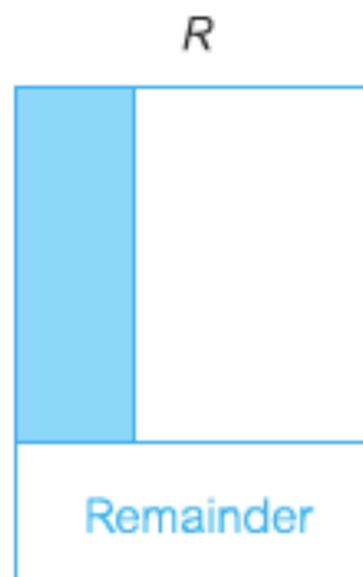
$T \bowtie_C U$

A	B	C
a	1	x
a	1	y
b	2	

(g) Natural join

(h) Semijoin

(i) Left Outer join



*V*

A	B
a	1
a	2
b	1
b	2
c	1

*W*

B
1
2

$V \div W$

A
a
b

(j) Division (shaded area)

Example of division

# Algebra Relacional

## União

$R \cup S$

A união de duas relações R e S define uma relação que contém todas os tuplos de R, ou S, ou ambos os R e S, os tuplos duplicados são eliminados. R e S devem ser compatíveis para a união.

Listar todas as cidades em que há uma filial ou um imóvel para alugar.

$$\pi_{\text{city}}(\text{Branch}) \cup \pi_{\text{city}}(\text{PropertyForRent})$$

city
London
Aberdeen
Glasgow
Bristol

# Algebra Relacional

## Diferença

**R - S**

A operação diferença define uma relação que consiste nos tuplos que são da relação R, mas não estão S. R e S devem ser compatíveis para a união.

Listar todas as cidades onde há uma filial, mas sem propriedades para alugar.

$$\pi_{\text{city}}(\text{Branch}) - \pi_{\text{city}}(\text{PropertyForRent})$$

city
Bristol

# Algebra Relacional

## Interseção

 $R \cap S$ 

A operação interseção define uma relação que consiste no conjunto de todos os tuplos que estão em ambos os R e S. R e S devem ser compatíveis para a união.

Listar todas as cidades onde há uma filial e pelo menos um imóvel para alugar.

$$\prod_{\text{city}} (\text{Branch}) \cap \prod_{\text{city}} (\text{PropertyForRent})$$

city
Aberdeen
London
Glasgow

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

# Algebra Relacional

## produto cartesiano

**R x S**

A operação produto cartesiano define uma relação que é a concatenação de cada tuplo de R relacionada com cada tuplo da relação S.

*Listar os nomes e comentários de todos os clientes que visualizaram um imóvel para alugar.*

$$\left( \prod_{\text{clientNo, fName, lName}} (\text{Client}) \right) \times \left( \prod_{\text{clientNo, propertyNo, comment}} (\text{Viewing}) \right)$$

# Algebra Relacional

client.clientNo	fName	IName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	

# Algebra Relacional

A relação contém informações a mais. Por exemplo, o primeiro tuplo contém diferentes valores *clientNo*. Para obter o resultado pretendido, precisamos realizar uma seleção nesta relação para extrair os tuplos onde **Client.clientNo = Viewing.clientNo**.

$$\sigma_{\text{Client.clientNo} = \text{Viewing.clientNo}} \left( \left( \prod_{\text{clientNo}, \text{fName}, \text{lName}} (\text{Client}) \right) \times \left( \prod_{\text{clientNo}, \text{propertyNo}, \text{comment}} (\text{Viewing}) \right) \right)$$

<b>client.clientNo</b>	<b>fName</b>	<b>lName</b>	<b>Viewing.clientNo</b>	<b>propertyNo</b>	<b>comment</b>
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

# Algebra Relacional

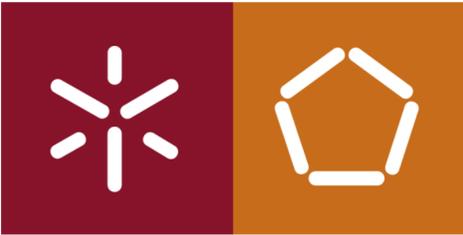
## Decomposição de operações complexas

$$\begin{aligned} \text{TempViewing}(\text{clientNo}, \text{propertyNo}, \text{comment}) &\leftarrow \Pi_{\text{clientNo}, \text{propertyNo}, \text{comment}}(\text{Viewing}) \\ \text{TempClient}(\text{clientNo}, \text{fName}, \text{lName}) &\leftarrow \Pi_{\text{clientNo}, \text{fName}, \text{lName}}(\text{Client}) \\ \text{Comment}(\text{clientNo}, \text{fName}, \text{lName}, \text{vclientNo}, \text{propertyNo}, \text{comment}) &\leftarrow \\ &\text{TempClient} \times \text{TempViewing} \\ \text{Result} &\leftarrow \sigma_{\text{clientNo} = \text{vclientNo}}(\text{Comment}) \end{aligned}$$

ou

$\rho_S(\mathbf{E})$  or  
 $\rho_{S(a_1, a_2, \dots, a_n)}(\mathbf{E})$

A mudança de nome fornece um novo nome S para a expressão de E, e, opcionalmente, os nomes dos atributos como  $a_1, a_2, \dots, a_n$ .



# Algebra Relacional

## Operador de JOIN (junção)

Existem várias formas de junção, cada uma com diferenças subtis, algumas mais úteis do que outras:

- Theta join
- Equijoin (uma forma particular de Theta join)
- Natural join
- Outer join
- Semijoin.

# Algebra Relacional

Theta join ( $\theta$ -join)

**$R \bowtie_F S$**

O operação Theta join define uma relação que contém os tuplos que satisfazem o predicado  $F$  a partir do produto cartesiano de  $R$  e  $S$ . O predicado  $F$  é da forma  $R.ai \theta S.bi$  onde  $\theta$  pode ser um dos operadores de comparação ( $<, \leq, >, \geq, =, \neq$ ).

$$R \bowtie_F S = \sigma_F(R \times S)$$

# Algebra Relacional

## Equijoin

Listar os nomes e comentários de todos os clientes que visualizaram um imóvel para alugar.

$$(\Pi_{\text{clientNo}, \text{fName}, \text{lName}}(\text{Client})) \bowtie_{\text{Client.clientNo} = \text{Viewing.clientNo}} (\Pi_{\text{clientNo}, \text{propertyNo}, \text{comment}}(\text{Viewing}))$$

$$\text{Result} \leftarrow \text{TempClient} \bowtie_{\text{TempClient.clientNo} = \text{TempViewing.clientNo}} \text{TempViewing}$$

# Algebra Relacional

## Natural join

 $R \bowtie S$ 

O Natural join é um Equijoin de duas relações R e S sobre todos os atributos comuns. Uma ocorrência comum de cada atributo é eliminada do resultado.

$$(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$$

Result  $\leftarrow$  TempClient  $\bowtie$  TempViewing

clientNo	fName	lName	propertyNo	comment
CR76	John	Kay	PG4	too remote
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR56	Aline	Stewart	PG36	
CR62	Mary	Tregear	PA14	no dining room

# Algebra Relacional

## Outer join

$$R \bowtie S$$

(left) Outer Join é uma junção em que os tuplos de R que não tem correspondência nos atributos comuns de S, são incluídos no resultado. Os valores em falta na segunda relação são apresentados a nulo.

*Produzir um relatório com o estado das propriedade visitadas.*

$$\left(\Pi_{\text{propertyNo, street, city}}(\text{PropertyForRent})\right) \bowtie \text{Viewing}$$

propertyNo	street	city	clientNo	viewDate	comment
PA14	16 Holhead	Aberdeen	CR56	24-May-04	too small
PA14	16 Holhead	Aberdeen	CR62	14-May-04	no dining room
PL94	6 Argyll St	London	null	null	null
PG4	6 Lawrence St	Glasgow	CR76	20-Apr-04	too remote
PG4	6 Lawrence St	Glasgow	CR56	26-May-04	
PG36	2 Manor Rd	Glasgow	CR56	28-Apr-04	
PG21	18 Dale Rd	Glasgow	null	null	null
PG16	5 Novar Dr	Glasgow	null	null	null

# Algebra Relacional

## Semijoin

 $R \triangleright_F S$ 

A operação Semijoin define uma relação que contém os tuplos de R que participam na junção de R com S.

$$R \triangleright_F S = \Pi_A(R \bowtie_F S)$$

Listar os detalhes completos de todos os funcionários que trabalham na filial de Glasgow.

Staff  $\triangleright_{\text{Staff.branchNo} = \text{Branch branchNo.}} (\sigma_{\text{city} = \text{'Glasgow'}} (\text{Branch}))$

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

# Algebra Relacional

## Operador divisão

**R ÷ S**

A operação Divisão define uma relação sobre o atributos C, que consiste no conjunto de tuplos de R que correspondem a combinação de cada tuplo em S.

$$T_1 \leftarrow \Pi_C(R)$$

$$T_2 \leftarrow \Pi_C((T_1 \times S) - R)$$

$$T \leftarrow T_1 - T_2$$

*Identificar todos os clientes que visualizaram todos os imóveis com três quartos.*

# Algebra Relacional

## Operador divisão

$$(\Pi_{\text{clientNo, propertyNo}}(\text{Viewing})) \div (\Pi_{\text{propertyNo}}(\sigma_{\text{rooms}=3}(\text{PropertyForRent})))$$

 $\Pi_{\text{clientNo, propertyNo}}(\text{Viewing})$ 

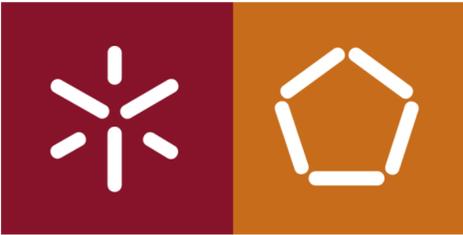
clientNo	propertyNo
CR56	PA14
CR76	PG4
CR56	PG4
CR62	PA14
CR56	PG36

 $\Pi_{\text{propertyNo}}(\sigma_{\text{rooms}=3}(\text{PropertyForRent}))$ 

propertyNo
PG4
PG36

**RESULT**

clientNo
CR56



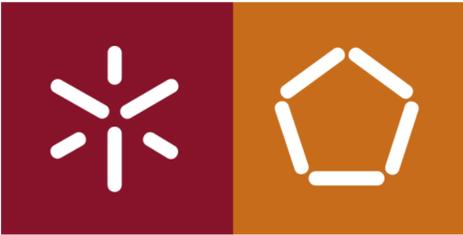
# Algebra Relacional

## Operações de Agregação e Agrupamento

### Operações de Agregação

 $\mathcal{S}_{AL}(R)$ 

Aplica-se a lista de funções de agregação, AL, para a relação R definindo uma relação sobre a lista agregada. AL contém um ou mais pares (<aggregate\_function>, <atributo>).



# Algebra Relacional

## Operações de Agregação e Agrupamento

As principais funções de agregação são:

- COUNT - retorna o número de valores do atributo associado.
- SUM - devolve a soma dos valores do atributo associado.
- AVG - devolve a média dos valores do atributo associado.
- MIN - retorna o menor valor no atributo associado.
- MAX - retorna o maior valor no atributo associado.

# Algebra Relacional

## Operações de Agregação

a) *Quantas propriedades para alugar custam mais de 350 por mês?*

$$\rho_R(\text{myCount}) \mathfrak{S} \text{ COUNT propertyNo } (\sigma_{\text{rent} > 350} (\text{PropertyForRent}))$$

b) *Encontre o mínimo, máximo e média dos salários dos funcionários.*

myCount
5

(a)

myMin	myMax	myAverage
9000	30000	17000

(b)

$$\rho_R(\text{myMin, myMax, myAverage}) \mathfrak{S} \text{ MIN salary, MAX salary, AVERAGE salary } (\text{Staff})$$

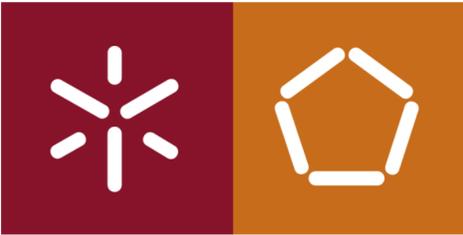
# Algebra Relacional

GA<sub>AL</sub>(R)

§

$\rho_R(\text{branchNo, myCount, mySum})_{\text{branchNo}} \int \text{COUNT staffNo, SUM salary} (\text{Staff})$

branchNo	myCount	mySum
B003	3	54000
B005	2	39000
B007	1	9000



# Algebra Relacional

- Ullman J.D. **Principles of Database and Knowledge-base Systems** Volume I. Rockville,MD: Computer Science Press, 1988.
- Molina H.G., Ullman J.D., Widom J. **Database Systems The Complete Book** Prentice Hall, 2nd Edition, 2008.
- Connolly, T., Begg, C., **Database Systems, A Practical Approach to Design, Implementation, and Management** , Addison-Wesley, 4a Edição, 2004