

Bases de Dados

PL09 – Álgebra Relacional

Docente: Diana Ferreira

Email: diana.ferreira@algoritmi.uminho.pt

Horário de Atendimento:

4^ª feira 10h-11h | DI 1.15



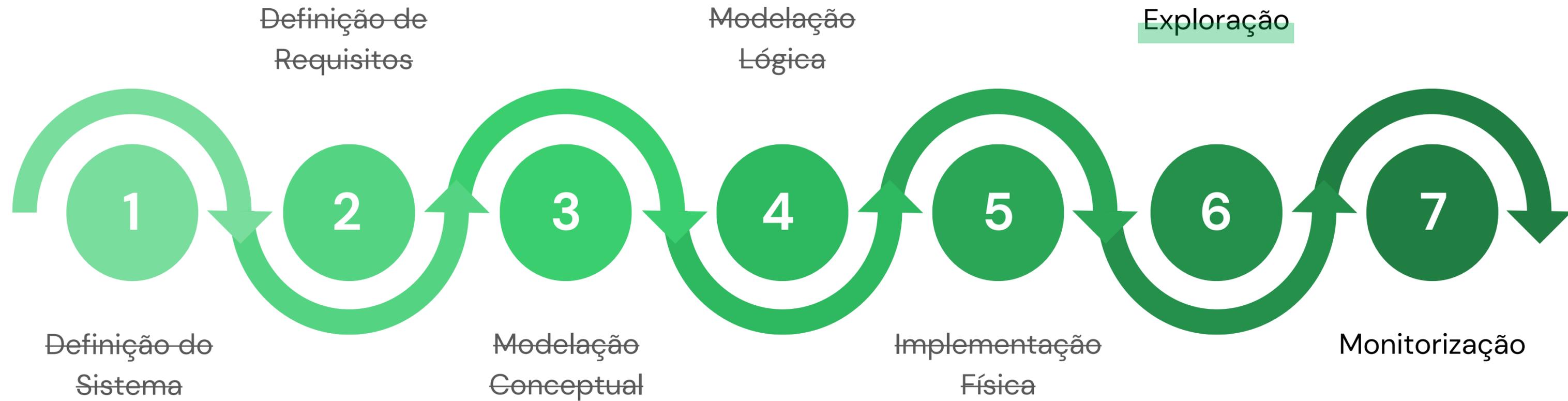
Sumário

1 Álgebra Relacional

Bibliografia:

- Connolly, T., Begg, C., Database Systems, A Practical Approach to Design, Implementation, and Management , Addison-Wesley, 4a Edição, 2004. **(Chapter 4 e 5)**
- Belo, O., "Bases de Dados Relacionais: Implementação com MySQL", FCA – Editora de Informática, 376p, Set 2021. ISBN: 978-972-722-921-5.

Ciclo de vida de um SBD



Álgebra Relacional

A Álgebra Relacional é uma linguagem teórica com operações que podem ser realizadas numa ou mais relações, definindo uma nova relação sem que as relações originais sejam modificadas.

Em álgebra relacional, podemos distinguir entre operações **unárias** (operações que ocorrem apenas sobre uma única relação) e **binárias** (operações que ocorrem sobre duas relações).

Álgebra Relacional

→ Operações

σ selecção

π projecção

\leftarrow atribuição

$\delta \rho$ renomeação

τ ordenação

γ agregação/agrupamento

\times produto cartesiano

\cup união

\cap intersecção

$-$ diferença

\bowtie junção

$\div /$ divisão/quociente

Álgebra Relacional

→ Operação de Seleção (σ)

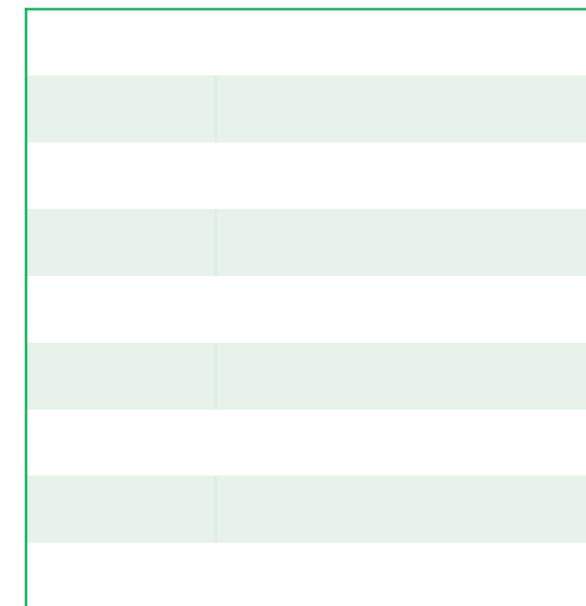
A operação de seleção, σ ("sigma"), é uma operação unária sobre uma dada relação R que dá origem a uma relação com esquema igual a R contendo apenas os registos que verifiquem a condição C de seleção. A condição de seleção pode envolver os operadores lógicos \wedge (AND), \vee (OR) e \neg (NOT) e os operadores de comparação $=$, \neq , $<$, \leq , $>$ ou \geq .

Notação:

$\sigma_C(R)$

Correspondência em SQL:

SELECT * FROM R WHERE C;



Operação de Seleção

Álgebra Relacional

→ Operação de Selecção (σ)

EXEMPLOS:

- Quais são os pacientes de Braga?

$\sigma_{localidade="Braga"} (pacientes)$

- Quais são os pacientes do sexo masculino solteiros?

$\sigma_{sexo="M" \wedge estado_civil="S"} (pacientes)$

- Quais são as consultas do paciente com o id '11114' cujo valor cobrado foi igual ou inferior a 20€?

$\sigma_{id_paciente=11114 \wedge preco \leq 20} (consultas)$

Álgebra Relacional

→ Operação de Projecção (π)

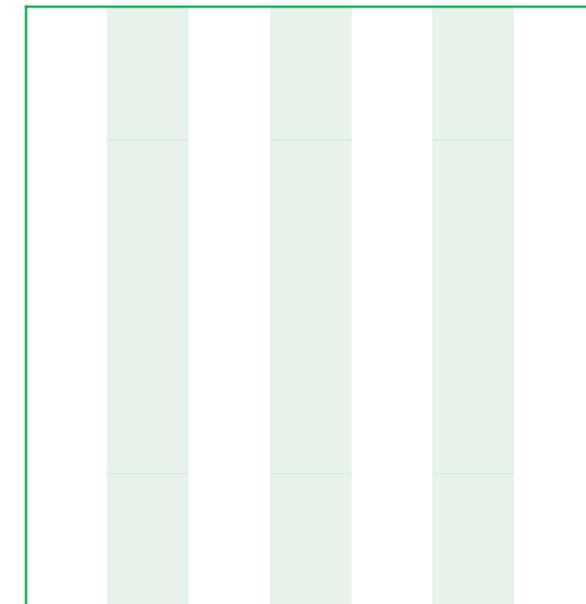
A operação de projecção, π ("pi"), é uma operação unária sobre uma dada relação R que dá origem a uma relação com um esquema contendo apenas a lista de atributos A_1, \dots, A_n especificados na operação de projecção, eliminando duplicados.

Notação:

$$\pi_{A_1, \dots, A_n}(R)$$

Correspondência em SQL:

***SELECT** A_1, \dots, A_n FROM R ;*



Operação de Projecção

Álgebra Relacional

→ Operação de Projecção (π)

EXEMPLOS:

- Quais são as localidades dos pacientes?

$\pi_{localidade} (pacientes)$

- Liste os fármacos e respetiva descrição.

$\pi_{nome, descricao} (farmacos)$

- Liste o preço de todas as consultas, indicando o número de episódio, a data de início e a data de fim.

$\pi_{preco, nr_episodio, hora_ini, hora_fim} (consultas)$

Álgebra Relacional

→ Operação de Selecção (σ) e Projecção (π)

EXEMPLOS:

- Qual é o nome e o sexo dos pacientes do Porto?

$\pi_{nome, sexo} (\sigma_{localidade="Porto"} (pacientes))$

- Qual é o nome dos procedimentos que possuem um valor superior a 15€.

$\pi_{des_procedimento} (\sigma_{preco>15} (procedimentos))$

- Liste os pacientes que não são do sexo masculino, indicando o seu nome, sexo e data de nascimento.

$\pi_{nome, sexo, data_nascimento} (\sigma_{sexo \neq "M"} (pacientes))$

Álgebra Relacional

→ Operação de Atribuição (\leftarrow)

A operação de atribuição, \leftarrow , é uma operação unária que permite atribuir o resultado de uma expressão de consulta em Álgebra Relacional a uma nova relação. As atribuições geralmente são usadas para aumentar a clareza, reduzindo uma consulta longa em várias etapas, descritas por linhas mais curtas. A atribuição armazena na relação R o *output* da expressão especificada.

Notação:

$$R \leftarrow [expressão]$$

Onde,

- R é o nome da nova relação;
- $[expressão]$ é uma expressão em algebra relacional.

Álgebra Relacional

→ Operação de Atribuição (\leftarrow)

EXEMPLOS:

- Qual é o nome e o sexo dos pacientes do Porto?

$$R_0 \leftarrow \sigma_{localidade="Porto"} (pacientes)$$

$$R \leftarrow \pi_{nome, sexo} (R_0)$$

- Qual é o nome dos procedimentos que possuem um valor superior a 15€.

$$R_0 \leftarrow (\sigma_{preco > 15} (procedimentos))$$

$$R \leftarrow \pi_{des_procedimento} (R_0)$$

- Liste os pacientes que não são do sexo masculino, indicando o seu nome, sexo e data de nascimento.

$$R_0 \leftarrow (\sigma_{sexo \neq "M"} (pacientes))$$

$$R \leftarrow \pi_{nome, sexo, data_nascimento} (R_0)$$

Álgebra Relacional

→ Operação de Renomeação (δ ρ)

A operação de renomeação ou mudança de nome, δ ("delta") ou ρ ("rho"), é uma operação unária sobre uma dada relação R que dá origem a uma relação com os atributos A_1, \dots, A_n renomeados para B_1, \dots, B_n .

Notação:

$$\rho_{(B_1, \dots, B_n)}(R)$$

$$\delta_{(A_1 \leftarrow B_1, \dots, A_n \leftarrow B_n)}(R)$$

Correspondência em SQL:

```
SELECT A1 AS B1, ..., An AS Bn FROM R;
```

Álgebra Relacional

→ Operação de Renomeação ($\delta \rho$)

EXEMPLOS:

- Renomear os atributos dos procedimentos para *codigo*, *descricao* e *valor*.

$$R \leftarrow \rho_{(codigo, descricao, valor)} (procedimentos)$$

$$R \leftarrow \delta_{cod_procedimento \leftarrow codigo, des_procedimento \leftarrow descricao, preco \leftarrow valor} (procedimentos)$$

- Renomear apenas os atributos *cod_procedimento* e *des_procedimento* dos procedimentos para *codigo* e *descricao*.

$$R \leftarrow \rho_{(codigo, descricao, preco)} (procedimentos)$$

$$R \leftarrow \delta_{cod_procedimento \leftarrow codigo, des_procedimento \leftarrow descricao} (procedimentos)$$

Álgebra Relacional

→ Operação de Renomeação, Seleção e Projeção

EXEMPLOS:

- Liste o nome e o ano de nascimento dos pacientes do sexo masculino.

$$R_0 \leftarrow \sigma_{sexo="M"} (pacientes)$$

$$R_1 \leftarrow \pi_{nome, YEAR(dta_nascimento)} (R_0)$$

$$R_2 \leftarrow \rho_{(nome, ano_nascimento)} (R_1)$$

ou

$$R \leftarrow \rho_{(nome, ano_nascimento)} (\pi_{nome, YEAR(dta_nascimento)} (\sigma_{sexo="M"} (pacientes)))$$



```
SELECT nome, YEAR(dta_nascimento) AS ano_nascimento
FROM pacientes WHERE sexo = "M";
```

Álgebra Relacional

→ Operação de Ordenação (τ)

A operação de ordenação, τ ("tau"), é uma operação unária sobre uma dada relação R que permite ordenar os tuplos segundo um ou mais atributos dessa mesma relação por ordem ascendente (ASC) ou descendente (DESC).

Notação:

$$\tau_{A_1[ASC|DESC], \dots, A_n[ASC|DESC]} (R)$$

Correspondência em SQL:

```
SELECT * FROM R ORDER BY A1 [ASC|DESC], ..., An [ASC|DESC];
```

Álgebra Relacional

→ Operação de Ordenação (τ)

EXEMPLOS:

- Liste os médicos por ordem ascendente de nome.

$$R \leftarrow \tau_{nome\ ASC} (medicos)$$

- Liste os pacientes do mais jovem para o mais velho.

$$R \leftarrow \tau_{data_nascimento\ DESC} (pacientes)$$

- Liste as consultas por ordem crescente de número de episódio e ordem decrescente de preço.

$$R \leftarrow \tau_{nr_episodio\ ASC, preco\ DESC} (consultas)$$

Álgebra Relacional

→ Operação de Ordenação com outras operações

EXEMPLOS:

- Liste os pacientes do mais jovem para o mais velho cujo ano de nascimento seja igual ou superior a 1990.

$$R \leftarrow \tau_{dta_nascimento\ DESC} (\sigma_{YEAR(dta_nascimento) \geq 1990} (\text{pacientes}))$$

OU

$$R_0 \leftarrow \sigma_{YEAR(dta_nascimento) \geq 1990} (\text{pacientes})$$

$$R_1 \leftarrow \tau_{dta_nascimento\ DESC} (R_0)$$

- Liste o número de episódio, a hora de início, a hora de fim e o preço das consultas por ordem crescente de número de episódio e ordem decrescente de preço.

$$R \leftarrow \tau_{nr_episodio\ ASC, preco\ DESC} (\pi_{nr_episodio, hora_ini, hora_fim, preco} (\text{consultas}))$$

OU

$$R_0 \leftarrow \pi_{nr_episodio, hora_ini, hora_fim, preco} (\text{consultas})$$

$$R \leftarrow \tau_{nr_episodio\ ASC, preco\ DESC} (R_0)$$

Álgebra Relacional

→ Operação de Agregação/Agrupamento (γ)

A operação de agregação/agrupamento, γ ("gama"), é uma operação unária que permite agregar os tuplos de uma dada relação R segundo um ou mais atributos (A_1, \dots, A_n) dessa mesma relação, com base num dado critério de agregação Op – COUNT(), SUM(), AVG(), MAX(), MIN(), etc e num ou mais atributos agrupadores $A_{agrupador}$.

Notação:

$$\gamma_{A_{agrupador_1}, A_{agrupador_2}, Op(A_1), \dots, Op(A_n)} (R)$$

Correspondência em SQL:

SELECT Op(A₁), ..., Op(A_n) FROM R GROUP BY A_{agrupador_1}, A_{agrupador_2};

Álgebra Relacional

→ Operação de Agregação/Agrupamento (γ)

EXEMPLOS:

- Qual é o valor médio das consultas?

$$R \leftarrow \gamma_{AVG(preco)} (consultas)$$

- Qual é o valor máximo e mínimo dos procedimentos?

$$R \leftarrow \gamma_{MAX(preco), MIN(preco)} (procedimentos)$$

- Indique o número de pacientes para cada sexo.

$$R \leftarrow \gamma_{sexo, COUNT(*)} (pacientes)$$

Álgebra Relacional

→ Operação de Agregação com outras operações

EXEMPLOS:

- Liste o valor máximo e mínimo de consultas por ano, ordenando-os do ano mais recente para o mais antigo.

$$R_0 \leftarrow \gamma_{YEAR(hora_ini), MAX(preco), MIN(preco)} (consultas)$$

$$R_1 \leftarrow \rho_{ano, max, min} (R_0)$$

$$R_2 \leftarrow \tau_{ano \text{ DESC}} (R_1)$$

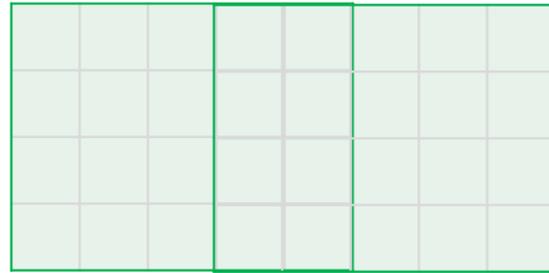
OU

$$R \leftarrow \tau_{ano \text{ DESC}} (\rho_{ano, max, min} (\gamma_{YEAR(hora_ini), MAX(preco), MIN(preco)} (consultas)))$$

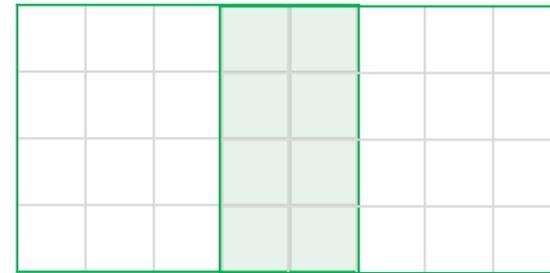
Álgebra Relacional

→ Operações de conjuntos

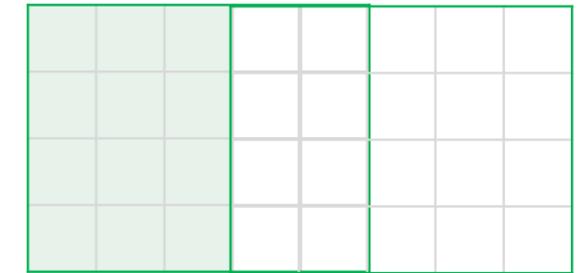
As operações sobre conjuntos unem duas relações, eliminando tuplas repetidas da relação.



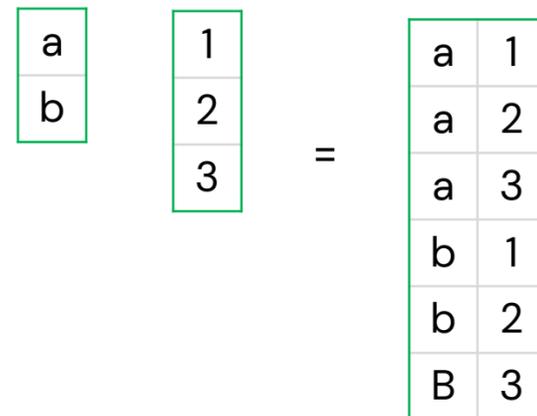
$R \cup S$ - união



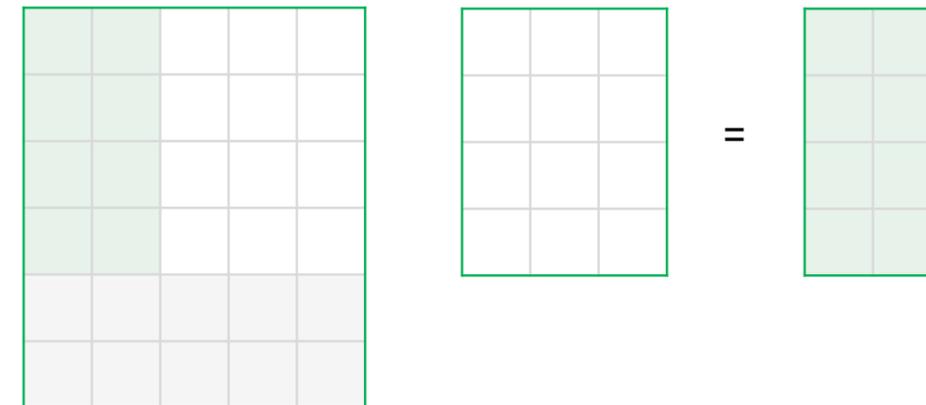
$R \cap S$ - intersecção



$R - S$ - diferença



$R \times S$ - produto cartesiano



$R \div S$ - divisão/quociente

Álgebra Relacional

→ Produto Cartesiano (\times)

O produto cartesiano, \times , é uma operação entre duas relações R e S que dá origem a uma relação que é a concatenação de cada tuplo de R relacionada com cada tuplo de S. O esquema da relação resultante contém todos os atributos de R e de S, apresentados pela ordem com que aparecem respetivamente em R e em S.

Notação:

$R \times S$

Correspondência em SQL:

*SELECT * FROM R, S;*
*SELECT * FROM R CROSS JOIN S;*

a	1	=	a	1
b	2		a	2
	3		a	3
			b	1
			b	2
			B	3

$R \times S$ – produto cartesiano

Álgebra Relacional

→ Produto Cartesiano (×)

EXEMPLOS:

- Combinação da relação "medicos" com a relação "especialidades" retorna:

$$R \leftarrow \text{medicos} \times \text{especialidades}$$

num_mec	nome_medico	cod_especialidade	cod_especialidade	des_especialidade
35	Artur Ferreira Soares	2221	2200	Anestesiologia
35	Artur Ferreira Soares	2221	2201	Angiologia
35	Artur Ferreira Soares	2221	2202	Cardiologia
35	Artur Ferreira Soares	2221	2203	Cirurgia Cardiotorácica
35	Artur Ferreira Soares	2221	2204	Cirurgia Geral
35	Artur Ferreira Soares	2221	2205	Cirurgia Vascular
35	Artur Ferreira Soares	2221	2206	Dermatologia
35	Artur Ferreira Soares	2221	2207	Endocrinologia
35	Artur Ferreira Soares	2221	2208	Estomatologia
35	Artur Ferreira Soares	2221	2209	Fisiatria
35	Artur Ferreira Soares	2221	2210	Gastrenterologia
35	Artur Ferreira Soares	2221	2211	Ginecologia
35	Artur Ferreira Soares	2221	2212	Medicina Interna
...

NOTA: Obtemos todas as combinações de medicos e de especialidades, o que não é informação muito útil.

Álgebra Relacional

→ Produto Cartesiano (\times)

EXEMPLOS:

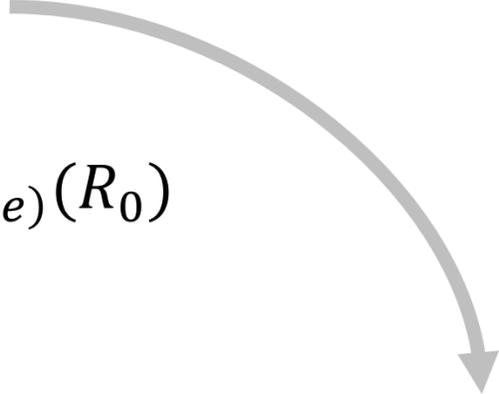
- Para listar os médicos e respetiva especialidade:

$$R_0 \leftarrow \text{medicos} \times \text{especialidades}$$

$$R_1 \leftarrow \rho_{(\text{num}_m, \text{nome}_m, \text{m}_c, \text{e}_c, \text{des}_e)}(R_0)$$

$$R_2 \leftarrow \sigma_{\text{m}_c = \text{e}_c}(R_1)$$

$$R_3 \leftarrow \pi_{\text{nome}_m, \text{des}_e}(R_2)$$



```
SELECT nome_medico, des_especialidade
FROM medicos e, especialidades e
WHERE e.cod_especialidade = e.cod_especialidade
```

Este exemplo corresponde a uma “junção natural”, conceito que introduziremos mais à frente.

Álgebra Relacional

→ Produto Cartesiano (×)

nome_medico	des_especialidade
Leonor Santos da Cunha	Cardiologia
Margarida Simões Branco	Cardiologia
José Fernando Noronha	Cardiologia
Cristiana Silva Pereira	Nefrologia
Eduardo Azevedo Castro	Neurologia
Filipa Silva Mendonça	Neurologia
Pedro Rodrigues Peixoto	Neurologia
Carlos André Cunha Fernandes	Neuropsicologia
Bruno Miguel Costa Ferreira	Neurocirurgia
Lígia Ribeiro Santos	Dermatologia
Bernardo Pereira Neto	Dermatologia
Marília Costa Azevedo	Psiquiatria
Álvaro Ribeiro Pascoal	Ortopedia
Leandro Correia Pereira	Ortopedia
Maria Leonor Dias Gomes	Obstetrícia
José Pereira Cabral	Nefrologia
Leonardo Morais Sarmento	Nefrologia
Patrícia Filipa Maia Almeida	Pediatria
Catarina Rodrigues Nogueira	Pediatria
Ricardo José Santos da Maia	Pediatria
Fernando Costa Tavares	Pediatria
Eduarda Maria da Silva Pascoal	Oftalmologia
Rita Gomes Teixeira	Oftalmologia
Artur Ferreira Soares	Oftalmologia

Álgebra Relacional

→ Operações de Junção

A operação de Junção é utilizada para combinação da informação contida entre duas ou mais tabelas. Uma junção pode ser definida como um produto cartesiano seguido por operações de seleção e de projeção.

\bowtie junção natural

$\bowtie_{A\theta B}$ teta-junção

$\bowtie_{A=B}$ equi-junção

\ltimes semi-junção

Inner joins

Outer joins

\ltimes junção externa esquerda

\rhd junção externa direita

\bowtie junção externa completa

Álgebra Relacional

→ Operação de Junção Natural (\bowtie)

A operação de Junção Natural, \bowtie , é uma operação entre duas relações R e S que permite inter-relacionar essas duas relações através de atributos que sejam comuns às duas relações e que possuam valores iguais. O esquema da relação resultante contém todos os atributos de ambas as relações – excluindo-se um dos atributos de junção.

Notação:

$R \bowtie S$

Correspondência em SQL:

*SELECT * FROM R NATURAL JOIN S;*

NOTA: Se as duas relações envolvidas numa operação de junção natural não possuírem qualquer atributo comum, então a operação de junção natural é equivalente a um produto cartesiano entre as duas relações.

Álgebra Relacional

→ Operação de Junção Natural (\bowtie)

EXEMPLOS:

- Quais são as especialidades exercidas pelos médicos?

$$R \leftarrow especialidades \bowtie medicos$$

- Liste as prescrições efetuadas e respetivos fármacos.

$$R \leftarrow farmacos \bowtie prescricoes$$

- Liste para cada prescricao, o nome do farmaco, a quantidade prescrita e a unidade.

$$R \leftarrow (\pi_{id_farmaco, nome}(farmaco)) \bowtie (\pi_{nr_episodio, id_paciente, id_medico, id_farmaco, quantidade, unidade}(prescricoes))$$

Álgebra Relacional

→ Operação de Teta-Junção ($\bowtie_{A\theta B}$)

A operação de Teta-Junção, $\bowtie_{A\theta B}$, é uma operação entre duas relações R e S que gera uma relação contendo todas tuplas resultantes do produto cartesiano ($R \times S$) que satisfaçam a expressão predicativa $A\theta B$, em que A é um qualquer atributo de R, B um qualquer atributo de S e θ um dos operadores de comparação =, \neq , <, \leq , > ou \geq .

Notação:

$$R \bowtie_{A\theta B} S = \sigma_{A\theta B}(R \times S)$$

Correspondência em SQL:

*SELECT * FROM R INNER JOIN S ON A θ B;*

Álgebra Relacional

→ Operação de Teta-Junção ($\bowtie_{A\theta B}$)

EXEMPLOS:

- Considere as seguintes relações:

Projetos

<u>Projeto</u>	Designação	Grau Exigido
001	Ensino à Distância	1
002	Avaliação Automática	3
003	Escola Feliz	2

Professores

<u>Professor</u>	Nome	Grau	Cidade
001	António Castro	1	Braga
002	José Silva	3	Porto
003	Cristina Campos	1	Vila Real

- Quais são os professores que podem ser responsáveis por projetos, sabendo-se que têm que ter um grau igual ou superior àquele que é exigido pelo projeto.

$$R \leftarrow \text{professores} \bowtie_{\text{professores.grau} \geq \text{projetos.grau_exigido}} \text{projetos}$$

Álgebra Relacional

→ Operação de Equi-Junção ($\bowtie_{A=B}$)

A operação de Equi-Junção, $\bowtie_{A=B}$, é um caso particular da teta-junção, que ocorre quando θ é substituído pelo operador de igualdade (=). A operação gera uma relação contendo todas tuplas resultantes do produto cartesiano ($R \times S$) que satisfaçam a expressão predicativa $A=B$, em que A é um qualquer atributo de R e B um qualquer atributo de S .

Notação:

$$R \bowtie_{A=B} S = \sigma_{A=B}(R \times S)$$

Correspondência em SQL:

*SELECT * FROM R INNER JOIN S ON R.A = S.B;*

*SELECT * FROM R INNER JOIN S USING (A);* Se os atributos de junção das duas tabelas tiverem o mesmo nome - A.

Álgebra Relacional

→ Operação de Equi-Junção ($\bowtie_{A=B}$)

EXEMPLOS:

- Quais são os médicos da especialidade de cardiologia?

$$R_0 \leftarrow \sigma_{des_especialidade="Cardiologia"}(especialidades)$$

$$R_1 \leftarrow R_0 \bowtie_{R_0.cod_especialidade=medicos.cod_especialidade} medicos$$

$$R_2 \leftarrow \pi_{num_mec,nome_medico,cod_especialidade}(R_1)$$

Álgebra Relacional

→ Operação de Junção Externa à Esquerda (\bowtie)

A operação de Junção Externa à Esquerda (*Outer Left Join*), \bowtie , integra na relação final todas as tuplas da relação à esquerda, mesmo quando estas não obedecem aos critérios de junção definidos. Ou seja, os tuplos de R que não têm correspondência nos atributos comuns de S são incluídos no resultado. Quando não existem valores correspondentes na segunda relação S, apresentam-se valores nulos (NULL).

Notação:

$R \bowtie S$

Correspondência em SQL:

*SELECT * FROM R LEFT JOIN S ON R.A = S.B;*

Álgebra Relacional

→ Operação de Junção Externa à Esquerda (\bowtie)

EXEMPLOS:

- Considere as seguintes relações:

Professores-Disciplinas

<u>Professor</u>	<u>Disciplina</u>
001	PRC
001	ANN
002	SBD
002	PRC

Professores

<u>Professor</u>	<u>Nome</u>	<u>Grau</u>	<u>Cidade</u>
001	António Castro	1	Braga
002	José Silva	3	Porto
003	Cristina Campos	1	Vila Real

- Qual o serviço lectivo que os professores têm (ou não)?

<u>Professor</u>	<u>Nome</u>	<u>Grau</u>	<u>Cidade</u>	<u>Disciplina</u>
001	António Castro	1	Braga	PRC
001	António Castro	1	Braga	ANN
002	José Silva	3	Porto	SBD
002	José Silva	3	Porto	PRC
003	Cristina Campos	1	Vila Real	<NULO>

$R \leftarrow \text{professores} \bowtie \text{professores_disciplinas}$

Quando não existem valores correspondentes apresentam-se nulos.

Registo sem equivalente na relação do lado direito.

Álgebra Relacional

→ Operação de Junção Externa à Direita (\bowtie)

A operação de Junção Externa à Direita (*Outer Right Join*), \bowtie , é semelhante Junção Externa à Esquerda, exceto que o tratamento das tabelas unidas é invertido. Ou seja, integra na relação final todas as tuplas da relação à direita, mesmo quando estas não obedecem aos critérios de junção definidos. Quando não existem valores correspondentes na primeira relação R , apresentam-se valores nulos (NULL).

Notação:

$R \bowtie S$

Correspondência em SQL:

*SELECT * FROM R RIGHT JOIN S ON R.A = S.B;*

Álgebra Relacional

→ Operação de Junção Externa à Direita (\bowtie)

EXEMPLOS:

- Considere as seguintes relações:

Professores-Disciplinas

Professor	Disciplina
001	PRC
001	ANN
002	SBD
002	PRC

Disciplinas

Disciplina	Designação	Tipo
PRC	Programação de Computadores	Semestral
ANN	Análise Numérica	Semestral
MAT	Matemática	Annual
SBD	Sistemas de Bases de Dados	Annual

- Quais as disciplinas que estão (ou não) a ser leccionadas?

Professor	Disciplina	Designação	Tipo
001	PRC	Programação de Computadores	Semestral
001	ANN	Análise Numérica	Semestral
<NULO>	MAT	Matemática	Annual
002	SBD	Sistemas de Bases de Dados	Annual
002	PRC	Programação de Computadores	Semestral

$R \leftarrow professores_disciplinas \bowtie disciplinas$

Álgebra Relacional

→ Operação de Junção Externa Completa (\bowtie)

A operação de Junção Externa Completa (*Full Outer Join*), \bowtie , integra na relação final todas as tuplas da relação à esquerda e da relação à direita, mesmo quando estas não obedecem aos critérios de junção definidos. Quando não existem valores correspondentes, apresentam-se valores nulos (NULL).

Notação:

$R \bowtie S$

Correspondência em SQL:

*SELECT * FROM R FULL
JOIN S ON R.A = S.B;*

O FULL JOIN não é suportado
pelo MySQL.



*SELECT * FROM R
LEFT JOIN S ON R.A = S.B
UNION
SELECT * FROM R
RIGHT JOIN S ON R.A = S.B*

Álgebra Relacional

→ Operação de Junção Externa Completa (\bowtie)

EXEMPLOS:

- Considere as seguintes relações:

Funcionários

Funcionários	Nome	Cidade
001	Fernando Costa	Braga
002	António Castro	Guimarães
003	Isabel Rodrigues	Viana

Professores

Professor	Nome	Grau	Cidade
001	António Castro	1	Braga
002	José Silva	3	Porto
003	Cristina Campos	1	Vila Real

- Quais os Funcionários e Professores que habitam (ou não) na mesma cidade?

Funcionários	Nome	Cidade	Professor	Nome	Grau
001	Fernando Costa	Braga	001	António Castro	1
002	António Castro	Guimarães	<NULO>	<NULO>	<NULO>
003	Isabel Rodrigues	Viana	<NULO>	<NULO>	<NULO>
<NULO>	<NULO>	Porto	002	José Silva	3
<NULO>	<NULO>	Vila Real	003	Cristina Campos	1

$R \leftarrow funcionarios \bowtie professores$

Álgebra Relacional

→ Operação de Semi-Junção (\bowtie)

A operação de Semi-Junção (*Semi Join*), \bowtie , sobre as relações R e S permite reduzir o número de tuplas envolvidos numa operação, com a aplicação de uma operação de projeção sobre a primeira relação (relação à esquerda) R envolvida na expressão. É semelhante à junção natural, mas o resultado da semi-junção é apenas o conjunto de todas as tuplas de uma tabela R onde uma ou mais correspondências são encontradas na segunda tabela S.

Notação:

$$R \bowtie S = \pi_{R_1, \dots, R_n}(R \bowtie S)$$

Correspondência em SQL:

SELECT R1, ..., Rn FROM R NATURAL JOIN S;

SELECT R1, ..., Rn FROM R WHERE A IN (SELECT A FROM S);

Álgebra Relacional

→ Operação de Semi-Junção (\bowtie)

EXEMPLOS:

- Quais os pacientes que foram consultados?

$$R_0 \leftarrow \delta_{id_paciente \leftarrow nr_sequencial}(consultas)$$

$$R \leftarrow Pacientes \bowtie R_0$$

- Quais os médicos que deram consultas?

$$R_0 \leftarrow \delta_{id_medico \leftarrow num_mec}(consultas)$$

$$R \leftarrow Medicos \bowtie R_0$$

Álgebra Relacional

→ Operação de União (U)

A operação de União, U , é uma operação entre duas relações compatíveis R e S que gera uma relação que contém todas as tuplas pertencentes a R , a S , ou a ambas, eliminando tuplas repetidas. Diz-se que duas relações são compatíveis se possuírem o mesmo grau (nº de colunas/atributos) e se os atributos/colunas correspondentes forem do mesmo domínio (tipo de dados).

Notação:

$R \cup S$

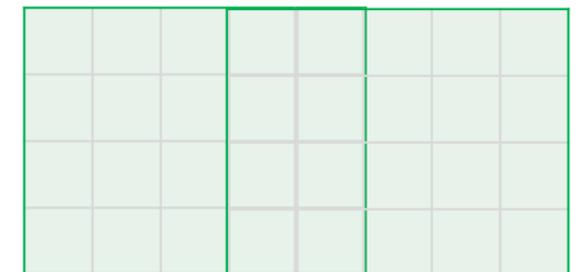
Correspondência em SQL:

*SELECT * FROM R*

UNION

*SELECT * FROM S;*

NOTA: Na qual as relações R e S possuem o mesmo nº de atributos e com domínios equivalentes.



$R \cup S$ - união

Álgebra Relacional

→ Operação de União (U)

EXEMPLOS:

- Liste os emails, tanto dos pacientes como dos funcionários, numa única relação.

$$R \leftarrow emails_{pac} \cup emails_{func}$$

- Liste os nomes dos pacientes de Braga ou dos pacientes que foram consultados ou ambos.

$$R_0 \leftarrow \sigma_{localidade="Braga"}(pacientes)$$

$$R_1 \leftarrow \pi_{nome}(R_0)$$

Pacientes de Braga

$$R_2 \leftarrow pacientes \bowtie_{pacientes.nr_sequencial=consultas.id_paciente} consultas$$

$$R_3 \leftarrow \pi_{nome}(R_2)$$

Pacientes que foram consultados

$$R_4 \leftarrow R_1 \cup R_3$$

Álgebra Relacional

→ Operação de Diferença (-)

A operação de Diferença, $-$, é uma operação entre duas relações compatíveis R e S que gera uma relação com esquema igual a R que contém todas as tuplas pertencentes a R , mas não pertencentes a S . Diz-se que duas relações são compatíveis se possuírem o mesmo grau (nº de colunas/atributos) e se os atributos/colunas correspondentes forem do mesmo domínio (tipo de dados).

Notação:

$R - S$

Correspondência em SQL:

```
SELECT * FROM R
EXCEPT/MINUS
SELECT * FROM S;
```

NOTA: Na qual as relações R e S são compatíveis.

O EXCEPT e o MINUS não são suportados pelo MySQL.



```
SELECT * FROM R
WHERE id NOT IN
(SELECT id FROM S);
```

NOTA: Na qual o atributo id existe em R e S com o mesmo domínio.

$R - S$ - diferença

Álgebra Relacional

→ Operação de Diferença (-)

EXEMPLOS:

- Quais os nomes dos médicos que nunca deram consultas?

$$R_0 \leftarrow \pi_{nome}(medicos)$$

Nomes dos Médicos

$$R_1 \leftarrow medicos \bowtie_{medicos.num_mec=consultas.id_medico} consultas$$

$$R_2 \leftarrow \pi_{nome}(R_1)$$

Nomes dos Médicos que deram consultas

$$R_3 \leftarrow R_0 - R_2$$

- Liste os nomes dos pacientes de Braga que não foram consultados.

$$R_0 \leftarrow \sigma_{localidade="Braga"}(pacientes)$$

$$R_1 \leftarrow \pi_{nome}(R_0)$$

Nomes dos Pacientes de Braga

$$R_2 \leftarrow pacientes \bowtie_{pacientes.nr_sequencial=consultas.id_paciente} consultas$$

$$R_3 \leftarrow \pi_{nome}(R_2)$$

Nomes dos pacientes que foram consultados

$$R_4 \leftarrow R_1 - R_3$$

Álgebra Relacional

→ Operação de Intersecção (\cap)

A operação de Intersecção, \cap , é uma operação entre duas relações compatíveis R e S que gera uma relação com esquema igual a R que contém todas as tuplas que pertencem simultaneamente a R e a S. Diz-se que duas relações são compatíveis se possuírem o mesmo grau (nº de colunas/atributos) e se os atributos/colunas correspondentes forem do mesmo domínio (tipo de dados).

Notação:

$$R \cap S = R - (R - S) \quad \text{Operação derivada}$$

Correspondência em SQL:

```
SELECT * FROM R
INTERSECT
SELECT * FROM S;
```

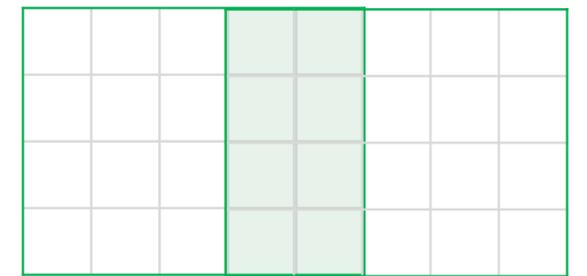
NOTA: Na qual as relações R e S são compatíveis.

O INTERSECT não é suportado pelo MySQL.



```
SELECT * FROM R
WHERE id IN
(SELECT id FROM S);
```

NOTA: Na qual o atributo id existe em R e S com o mesmo domínio.



$R \cap S$ - intersecção

```
SELECT * FROM R
WHERE EXISTS
(SELECT * FROM S
WHERE R.id = S.id);
```

Álgebra Relacional

→ Operação de Intersecção (\cap)

EXEMPLOS:

- Quais os nomes dos médicos que já deram consultas?

$$R_0 \leftarrow \pi_{nome}(medicos)$$

Nomes dos Médicos

$$R_1 \leftarrow medicos \bowtie_{medicos.num_mec=consultas.id_medico} consultas$$

$$R_2 \leftarrow \pi_{nome}(R_1)$$

Nomes dos Médicos que deram consultas

$$R_3 \leftarrow R_0 \cap R_2$$

- Liste os nomes dos pacientes de Braga que foram consultados.

$$R_0 \leftarrow \sigma_{localidade="Braga"}(pacientes)$$

$$R_1 \leftarrow \pi_{nome}(R_0)$$

Nomes dos Pacientes de Braga

$$R_2 \leftarrow pacientes \bowtie_{pacientes.nr_sequencial=consultas.id_paciente} consultas$$

$$R_3 \leftarrow \pi_{nome}(R_2)$$

Nomes dos pacientes que foram consultados

$$R_4 \leftarrow R_1 \cap R_3$$

Álgebra Relacional

→ Operação de Divisão (\div /)

A operação de Divisão, \div /, é uma operação entre duas relações R e S, na qual os atributos de S devem constituir um subconjunto dos atributos de R. Esta operação gera uma relação com esquema igual a todos os atributos de R que não são de S, através da seleção de tuplas da relação R que façam referência a todas as tuplas da relação S.

Notação:

$$R \div S = \pi_A(R) - \pi_A((\pi_A(R) \times S) - R)$$

Correspondência em SQL:

A divisão não é suportada por implementações SQL. No entanto, pode ser representada usando outras operações (como not exists, not in, count, etc.)



$R \div S$ - divisão/quociente

Álgebra Relacional

→ Operação de Divisão (\div /)

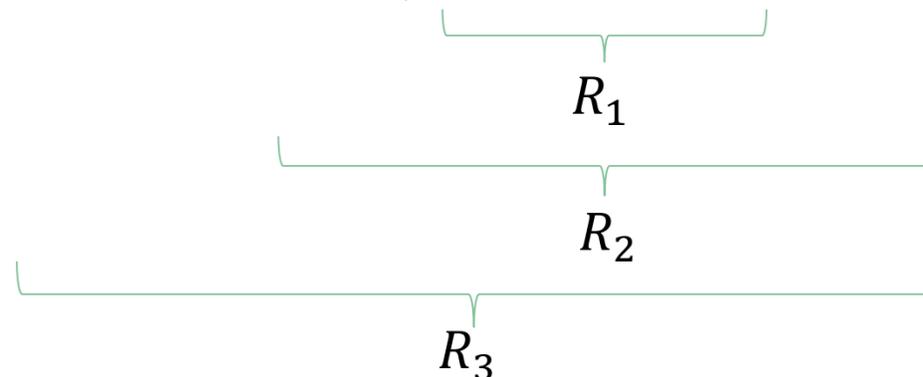
Dadas duas relações $R(x,y)$, $S(y)$.

$R(x,y) / S(y)$ significa que queremos retornar todos os valores distintos de x de R que estão associados a todos os valores de y de S .

Passos:

- R_1 - Descubra todas as combinações possíveis de $S(y)$ com $R(x)$ calculando $R(x) \times S(y)$
- R_2 - Subtraia a relação $R(x,y)$ por R_1
- R_3 - Em R_2 obtivemos os tuplos de R que não estão associados a todos os valores em $S(y)$; Portanto $R(x) - R_2(x)$ retorna os x que estão associados a todos os valores em S .

$$R \div S = \pi_x(R) - \pi_x((\pi_x(R) \times S) - R)$$



Álgebra Relacional

→ Operação de Divisão (\div)

EXEMPLOS:

- Quais os nomes dos pacientes que já foram atendidos por todos os médicos de Obstetrícia?

1º - retornar os médicos que exercem a especialidade de obstetrícia

$$R_1 \leftarrow \text{medicos} \bowtie \text{especialidades}$$

$$R_2 \leftarrow \sigma_{\text{des_especialidade}=\text{"Obstetrícia"}}(R_1)$$

$$R_3 \leftarrow \rho_{\text{id_medico}}(\pi_{\text{num_mec}})$$

2º - retornar as consultas apenas com o id_paciente e com o id_medico

$$R_4 \leftarrow \pi_{\text{id_paciente}, \text{id_medico}}(\text{consultas})$$

3º - aplicar a divisão

$$R_5 \leftarrow R_4 \div R_3$$

4º - retornar o nome de cada id do paciente obtido

$$R_6 \leftarrow \text{pacientes} \bowtie R_5$$

$$R_7 \leftarrow \pi_{\text{nome}}(R_6)$$

Álgebra Relacional

→ Operação de Divisão (\div)

- Quais os nomes dos pacientes que já foram atendidos por todos os médicos de Obstetrícia?

1º - NOT EXISTS + NOT IN

```
SELECT nome FROM pacientes p WHERE NOT EXISTS
  (SELECT * FROM medicos m NATURAL JOIN especialidades e WHERE e.des_especialidade='Obstetrícia'
   AND m.num_mec NOT IN (SELECT id_medico FROM consultas co WHERE p.nr_sequencial=co.id_paciente)
  );
```

2º - NOT EXISTS + NOT EXISTS

```
SELECT nome FROM pacientes p WHERE NOT EXISTS
  (SELECT * FROM medicos m NATURAL JOIN especialidades e WHERE e.des_especialidade='Obstetrícia' AND
   NOT EXISTS (SELECT id_medico FROM consultas co WHERE p.nr_sequencial=co.id_paciente AND
   m.num_mec=co.id_medico));
```

3º - COUNT

```
SELECT p.nome FROM pacientes p INNER JOIN consultas c ON c.id_paciente=p.nr_sequencial WHERE
c.id_medico IN (SELECT num_mec FROM medicos m NATURAL JOIN especialidades e WHERE
e.des_especialidade='Obstetrícia') group by p.nome having count(c.id_medico) = (SELECT COUNT(*) FROM
medicos me NATURAL JOIN especialidades es WHERE es.des_especialidade='Obstetrícia');
```

Próxima aula: Exploração da BD

