

# Banco de Dados I

Prof. Diego Buchinger  
diego.buchinger@outlook.com  
diego.buchinger@udesc.br

Profa. Rebeca Schroeder Freitas  
Prof. Fabiano Baldo

---

# Álgebra Relacional

---

## **Categorias de linguagens de manipulação:**

- Formais: Álgebra Relacional & Cálculo Relacional
- Comerciais: SQL (baseadas nas formais)

## **Linguagens formais – Características**

- Orientadas a conjuntos
- Linguagens de base – devem ter no mínimo um poder de expressão equivalente ao de uma linguagem formal
- Fechamento – resultados de consultas são relações

## Linguagem procedural

- Expressões definem uma execução sequencial de operadores
- A execução de cada operador produz uma relação

## Classificação de operadores

### ➤ Fundamentais

- Unários: seleção, projeção
- Binários: produto cartesiano, união e diferença

### ➤ Derivados

- Binários: intersecção, junção e divisão

### ➤ Especiais: renomeação, atribuição e alteração

## Exemplo de Esquema Relacional

Ambulatórios ( nroa, andar, capacidade )

Médicos ( codm, CPF, nome, idade, cidade,  
especialidade, nroa )

Pacientes ( codp, CPF, nome, idade, cidade)

Consultas ( codm, codp, data, hora, doença)

Funcionários ( codf, CPF, nome, idade, cidade, salário )

## Ferramenta online

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

base do esquema acima na página

# Seleção

---

- Retorna **tuplas** que satisfazem um **predicado**
- Notação:  $\sigma_{predicado} (relação)$   
(sigma)
- Operadores de comparação:  $=, <, <=, >, >=, \neq$
- Operadores lógicos:  $\wedge$  (and)  $\vee$  (or)  $\neg$  (not)
- Exemplo:  $\sigma_{z \geq 2} (R)$

*R*

x	y	z
1	1	1
2	2	2
2	3	3

*resultado*

x	y	z
2	2	2
2	3	3

# Seleção

---

## Resolva:

- Buscar os dados dos pacientes que estão com ‘sarampo’
- Buscar os dados dos médicos ‘Ortopedista’ com mais de 55 anos
- Buscar os dados de todas as consultas, exceto aquelas marcadas para os médicos CRM 102401 e 100985
- Buscar os dados dos ambulatórios do quarto andar. Estes ambulatórios devem ter capacidade superior ou igual a 50 ou o número do ambulatório deve ser superior a 410.

# Projeção

---

- Retorna um ou mais atributos de interesse
- Notação:  $\pi_{lista\_nomes\_atributos} (relação)$   
(pi)
- Elimina automaticamente duplicatas
- Exemplo:  $\pi_{x,y} (R)$

*R*

x	y	z
1	1	1
2	2	2
2	2	3

*resultado*

x	y
1	1
2	2



# Projeção

---

## Resolva:

- Buscar nome e especialidade de todos os médicos
- Buscar o número dos ambulatórios do terceiro andar
- Buscar o código dos médicos e as datas das consultas para os pacientes com código 00003 e 00008
- Buscar os números dos ambulatórios com capacidade igual ou superior a 20, exceto aqueles do segundo andar

# Produto Cartesiano

- Retorna todas as combinações de tuplas de duas relações  $R_1$  e  $R_2$
- Grau do resultado:  $grau(R_1) + grau(R_2)$
- Cardinalidade do resultado:  $card(R_1) * card(R_2)$
- Notação:  $relação1 \times relação2$

(cross joint)

- Exemplo:  $(R1 \times R2)$

$R1$

x	y	z
1	1	1
2	2	2
3	3	3

$R2$

w	y
1	1
2	2

*resultado*

x	R1y	z	w	R2y
1	1	1	1	1
1	1	1	2	2
2	2	2	1	1
2	2	2	2	2
3	3	3	1	1
3	3	3	2	2

# Produto Cartesiano

---

**Resolva:**

- Buscar o **nome** dos médicos que têm/tiveram consulta marcada e as **datas** das suas consultas
- Buscar o **nº** e a **capacidade** dos ambulatórios do 3º andar e o **nome** dos médicos ‘Ortopedista’ que atendem neles
- Buscar entre as consultas que estão/foram marcadas para o período da manhã (07:00-12:00) a partir do ano de 2010, o **nome** do médico, paciente e a **data** da consulta
- Buscar o **nome** e **salário** dos funcionários de ‘Florianopolis’ que estão internados como pacientes e têm/tiveram consulta marcada com ‘Psiquiatra’

# Atribuição

---

- Armazena o resultado de uma expressão algébrica em uma **variável de relação**
- Possibilita processamento de consulta por etapas
- Notação:  $\text{nomeVariavel} = \text{ExpressaoAlgebrica}$   
(assignment)

- Exemplo:

$$R1 = \pi_{\text{codm}, \text{data}} (\text{Consultas})$$

$$R2 = \pi_{\text{codm}, \text{nome}} (\text{Medicos})$$

$$\pi_{\text{codm}, \text{data}} (\sigma_{\text{Consultas.codm}=\text{Medicos.codm}} (R1 \times R2))$$

# Atribuição

---

**Resolva:**

- Buscar o **nome** e a **idade** dos Pacientes que são de ‘São Bento do Sul’ utilizando atribuição
- Buscar os pacientes que tiveram consultas agendadas no período da tarde (13:00 e 19:00), mostrando o **nome** do paciente, o **nome** do médico que lhe atendeu e o **número** do ambulatório onde este médico atua.

- **Antecipação de seleções**
  - Realizar filtragens horizontais
  - Filtrar colunas o mais cedo possível

$\pi$  Pacientes.codp, cpf, nome, data, hora (Pacientes  $\times$  Consultas)

VS

$(\pi \text{ codp, cpf, nome (Pacientes)}) \times (\pi \text{ codp, data, hora (Consultas)})$

- **Antecipação de predicados**

- Realizar filtragens verticais

- Filtrar linhas o mais cedo possível

$$\sigma \text{ Consultas.codp} = \text{Pacientes.codp} \wedge \text{Consultas.data} < '2010/01/01'$$
$$(\text{Consultas} \times \text{Pacientes})$$
$$\sigma \text{ Consultas.codp} = \text{Pacientes.codp} ((\sigma \text{ data} < '2010/01/01'$$
$$(\text{Consultas})) \times \text{Pacientes})$$

- **Identificar sub expressões (comuns)**
  - Evitar repetir execução múltiplas de mesma operação  
Usar variável de relação para armazenar resultados  
[mais comum para múltiplas consultas]
  - Expressão fica visualmente mais legível  
[ (-) pode usar espaço extra de forma desnecessária]

$$R1 = \sigma \text{ Pacientes.codp} = \text{Consultas.codp} ((\pi \text{ codp, nome, idade Pacientes}) \\ \times (\pi \text{ codm, codp, data Consultas}))$$
$$R2 = \sigma \text{ Consultas.codm} = \text{Medicos.codm} (R1 \times \pi \text{ codm, nome, idade Medicos}) \\ \pi \text{ Pacientes.nome, Pacientes.idade, Medicos.nome, Medicos.idade (R2)}$$



# Renomeação

- Altera o nome de uma relação ou de seus atributos
- Notação:  $(rho)$ 
  - renomear relação:  $\rho_{novoNomeRelação} (Relação)$
  - renomear colunas:  $\rho_{novo1 \leftarrow orig1, novo2 \leftarrow orig2} (Relação)$
- Exemplo:  $(R \times \rho_{N1}(R))$   $\rho_{a \leftarrow x, b \leftarrow y}(R)$

$R$

x	y
1	2
3	4

*resultado*

R.x	R.y	N1.x	N1.y
1	2	1	2
1	2	3	4
3	4	1	2
3	4	3	4

R.a	R.b
1	2
3	4

*resultado*

# Renomeação

---

## Resolva:

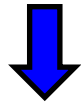
- Apresente a relação de todos os funcionários que são de ‘Joinville’ simplificando o nome da relação para F (ao invés de Funcionarios – ex: F.codf, F.cpf, F.nome)
- Apresentar uma listagem das consultas mostrando apenas o código do médico, o código do paciente e a doença, com o cabeçalho: **medico, paciente, problema**

# Otimização usando Renomeação

---

**Resolva:**

- Buscar o número dos ambulatórios onde pelo menos dois médicos de Joinville dão atendimento:

$$\pi M.nroa (\sigma M.codm \neq Medicos.codm \wedge M.nroa = Medicos.nroa (\sigma cidade='Joinville' (Medicos) \times \sigma cidade='Joinville' (\rho M (Medicos))) )$$


melhorando...

$$R1 = \pi codm, nroa (\sigma cidade='Joinville' (\rho M (Medicos)))$$
$$\pi M.nroa (\sigma M.codm \neq R2.codm \wedge M.nroa = R2.nroa (R1 \times (\rho R2 (R1))) )$$

- Nota: e se fosse preciso mostrar os nomes dos médicos?

## Resolva:

- Buscar o número dos ambulatórios com capacidade superior à capacidade do ambulatório de número 211
- Buscar o **nome** e o **CPF** dos funcionários que recebem salários iguais ou inferiores ao salário do funcionário com CPF '05432105412'
- Buscar pares de **nomes** de médicos diferentes que têm consultas marcadas nas mesmas datas. Apresentar os pares de nomes e a data em comum.

# União, Diferença e Intersecção

---

- Operam somente sobre duas relações compatíveis
  - grau ( $R_1$ ) = grau ( $R_2$ )
  - domínio atributo  $a_i$  de  $R_1$  = domínio atributo  $b_i$  de  $R_2$
- Resultado:
  - Grau: grau ( $R_1$ ) [= grau ( $R_2$ )]
  - Atributos: nomes dos atributos de  $R_1$  (relação esquerda)

# União

---

- Retorna a união das tuplas de duas relações
- Elimina duplicatas automaticamente
- Notação:  $relacao_1 \cup relacao_2$
- Exemplo:  $(R_1 \cup R_2)$

$R_1$

x	y
1	2
3	4
2	4

$R_2$

x	y
1	3
3	4
1	2
3	5

$resultado$

x	y
1	2
3	4
2	4
1	3
3	5

# Diferença

---

- Retorna as tuplas que estão em  $R_1$  e não estão em  $R_2$
- Notação:  $relacao_1 - relacao_2$
- Exemplo:  $(R_1 - R_2)$

$R_1$	<b>x</b>	<b>y</b>	$R_2$	<b>x</b>	<b>y</b>	$resultado$
	1	2		1	3	
	3	4		3	4	
	2	4		1	2	
				3	5	

# Intersecção

---

- Retorna as tuplas comuns entre  $R_1$  e  $R_2$
- Notação:  $relacao_1 \cap relacao_2$
- Exemplo:  $(R_1 \cap R_2)$

$R_1$	<b>x</b>	<b>y</b>	$R_2$	<b>x</b>	<b>y</b>	$resultado$
	1	2		1	3	
	3	4		3	4	
	2	4		1	2	
				3	5	



# União, Diferença e Intersecção

---

**Resolva:** (usando união, diferença e intersecção)

- Buscar **nome** e **CPF** dos médicos e dos pacientes cadastrados no hospital.
- Buscar **nome** e **CPF** dos funcionários que já foram pacientes na clínica.
- Buscar **nome**, **CPF** e **idade** dos médicos, pacientes e funcionários que residem em 'Joinville'.
- Buscar **número** dos ambulatórios onde nenhum médico dá atendimento.
- Buscar **nome** e **CPF** dos funcionários que recebem salários abaixo de R\$ 1.200 e não são pacientes.

# Junção (*Join*)

---

- Retorna a combinação de tuplas de duas relações  $R_1$  e  $R_2$  que satisfazem um predicado.
- Tipos de junção:
  - *Junção Natural*
  - *Junções Externas*
    - *Junção externa à esquerda*
    - *Junção externa à direita*
    - *Junção externa completa*
  - *Semi-Junção*
  - *Anti-Junção*

# Junção Natural (*Natural Join*)

---

- Junção em que a igualdade é predefinida entre todos os atributos que apresentam o mesmo nome nas relações  
(similar a um produto cartesiano condicional)

- Notação:  $relacao_1 \bowtie_{(condição)} relacao_2$

- Exemplo:  $(R_1 \bowtie_{(R1.y = R2.y)} R_2)$

$R_1$

x	y	z
1	1	1
1	1	2
2	2	3

$R_2$

w	y
7	1
4	2

*resultado*

x	y	z	w
1	1	1	7
1	1	2	7
2	2	3	4

# Junção Natural (*Natural Join*)

- Exemplo:  $(R_1 \bowtie R_2)$

 $R_1$ 

x	y	z
1	1	1
1	1	2
2	2	3

 $R_2$ 

x	y	w
1	1	3
2	4	2

*resultado*

x	y	z	w
1	1	1	3
1	1	2	3

- Exemplo:  $(R_1 \bowtie R_2) = (R_1 \times R_2)$

 $R_1$ 

x	y	z
1	1	1
1	2	5

 $R_2$ 

w	t
7	1
4	2

*resultado*

x	y	z	w	t
1	1	1	7	1
1	1	1	4	2
1	2	5	7	1
1	2	5	4	2

# Junção Natural

---

## Resolva:

- Buscar o **número** e a **capacidade** dos ambulatórios do terceiro andar e o **nome** dos médicos que atendem neles.
- Buscar o **nome** e o **salário** dos funcionários de ‘Florianopolis’ e ‘Joinville’ que tiveram consulta marcada no ano de 2015.
- Buscar o **número** e o **andar** dos ambulatórios onde nenhum médico atende.
- Buscar o **número** dos ambulatórios que estão no mesmo andar do ambulatório 311 e possuem capacidade superior.

# Junções Externas (*Outer Joins*)

---

- Junção em que as tuplas de uma ou ambas as relações que não são combinadas são mesmo assim preservadas
- Tipos:
  - Junção Externa à Esquerda (*left [outer] join*)
    - apenas tuplas da relação à esquerda são preservadas
    - Notação:  $relacao_1 \bowtie_{(condição)} relacao_2$
  - Junção Externa à Direita (*right [outer] join*)
    - apenas tuplas da relação à direita são preservadas
    - Notação:  $relacao_1 \bowtie_{(condição)} relacao_2$
  - Junção Externa Completa (*full [outer] join*)
    - tuplas de ambas as relações são preservadas
    - Notação:  $relacao_1 \bowtie_{(condição)} relacao_2$

# Junções Externas (*Outer Joins*)

---

- Exemplos:

$R_1$	x	y	z
	1	1	1
	2	1	2
	3	3	3
	5	5	5

$R_2$	x	a	b
	1	7	3
	2	4	2
	4	4	4

# Junções Externas (*Outer Joins*)

- Exemplos:

$$(R_1 \bowtie R_2)$$

x	y	z	a	b
1	1	1	7	3
2	1	2	4	2
3	3	3	-	-
5	5	5	-	-
4	-	-	4	4

$$(R_1 \bowtie_{(R1.x = R2.x)} R_2)$$

x	y	z	a	b
1	1	1	7	3
2	1	2	4	2
3	3	3	-	-
5	5	5	-	-

$$(R_1 \bowtie R_2)$$

x	y	z	a	b
1	1	1	7	3
2	1	2	4	2
4	-	-	4	4



## Junções Externas (*Outer Joins*)

---

### Resolva:

- Buscar o **nome**, **especialidade** de todos os médicos e, para aqueles que têm consultas marcadas, mostrar a **data** de suas consultas também.
- Buscar os **números** de todos os ambulatórios e, para aqueles nos quais existe algum médico que dá atendimento, exibir o seu **código** e **nome**
- Mostrar em uma relação o **CPF** e o **nome** de todos os pacientes e funcionários. Para os funcionários, listar também o **salário**. Quando um funcionário for também paciente mostrar apenas uma vez o **CPF**, **nome** e **salário**.

## Semi-Junção (*Semi Join*)

---

- Versão de junção semelhante à Junção Natural. Difere apenas no fato de que são preservadas apenas as colunas da relação à esquerda (*left semi join*) ou da relação à direita (*right semi join*)
- Notação:  
$$relacao_1 \bowtie relacao_2$$
$$relacao_1 \bowtie relacao_2$$

# Semi-Junção (*Semi Join*)

- Exemplos:

Nome	Id	Dept
Alfredo	3415	Finanças
Beto	2241	Vendas
Carla	3401	Finanças
Djenifer	2202	Produção

Dept	Gerente
Produção	Zelia
Vendas	Yury
Admin	Willy

$$(R_1 \bowtie R_2)$$

Nome	Id	Depto
Beto	2241	Vendas
Djenifer	2202	Produção

$$(R_1 \bowtie R_2)$$

Dept	Gerente
Produção	Zelia
Vendas	Yury

## Anti-Junção (*Anti Join*)

---

- Junção similar a Junção Natural, mas preservam-se apenas as tuplas da relação da esquerda que possuem valor(es) na(s) coluna(s) comum(s) que não aparecem na relação da direita.
- Notação:  $relacao_1 \triangleright relacao_2$
- Pode ser escrito como:

$$R \triangleright S = R - (R \bowtie S)$$

# Anti-Junção (*Anti Join*)

---

- Exemplos:

Nome	Id	Dept
Alfredo	3415	Finanças
Beto	2241	Vendas
Carla	3401	Finanças
Djenifer	2202	Produção

Dept	Gerente
Produção	Zelia
Vendas	Yury
Admin	Willy

$$(R_1 \triangleright R_2)$$

Nome	Id	Depto
Alfredo	3415	Finanças
Carla	3401	Finanças

$$(R_2 \triangleright R_1)$$

Dept	Gerente
Admin	Willy

# Ordenar (*Order by*)

---

- Ordena os resultados baseado em uma ou mais colunas em ordem crescente ou decrescente

*tau*

- Notação:  $\tau_{[coluna\ ordem]}(relacao_1)$
- Ordem pode assumir valores: asc, desc
- A lista de *coluna ordem* deve ser separada por vírgula

$R_1$

x	y	z
1	2	a
4	1	b
2	3	c
2	5	c

$\tau_{x\ desc, y\ asc}(R_1)$

*Resultado*

x	y	z
4	1	b
2	3	c
2	5	c
1	2	a

# Agrupar (*Group by*)

---

- Agrupar os resultados e permite a realização de alguma função sobre o agrupamento (contagem, soma, ...)

*gamma*

- Notação:  $\gamma_{[função(coluna) \rightarrow novo\_nome]} (relacao_1)$
- Função pode ser: count( ), sum( ), avg( ), min( ), max( )
- A lista de agrupamentos deve ser separada por vírgula

$R_1$

x	y	z
1	2	a
4	1	b
2	3	c
2	5	c

$\gamma_{count(x) \rightarrow qtd, sum(y) \rightarrow somaY} (R_1)$

*Resultado*

qtd	somaY
4	11

# Semi-Junção, Anti-Junção, Ordenação e Agrupamento

---

## Resolva:

- Listar **CPF**, **nome** e **cidade** dos pacientes que tiveram ‘sarampo’ usando Semi-Junção.
- Listar o **nome** e a **idade** dos funcionários que nunca foram pacientes no hospital (ou seja, não possuem registro como pacientes) ordenado pela idade (crescente) e caso a idade seja igual ordenar pelo nome (decrescente).
- Calcular qual é a **capacidade máxima** do hospital considerando os ambulatorios registrados.
- Contar quantos ambulatorios não possuem um médico atendendo.



# Divisão

---

- Operação entre duas relações
  - Dividendo (grau  $m + n$ )
  - Divisor (grau  $n$ )
- Grau “n”: atributos de mesmo nome nas relações
- Grau “m” ou quociente: atributos da relação dividendo cujos valores associam-se com TODOS os valores da relação divisor
- Notação:  $relacao_1 \div relacao_2$

**OBS:** No RELAX, procure deixar as colunas em comum por último.

# Divisão

- Exemplos:

 $R_1$ 

x	y	z
1	1	1
1	2	1
2	1	1
2	2	2
3	1	3

 $R_2$ 

z
1

 $R_3$ 

y	z
1	1

 $R_4$ 

y
1
2

$R_1 \div R_2$

x	y
1	1
1	2
2	1

$R_1 \div R_3$

x
1
2

$R_1 \div R_4$

x	z
1	1

**OBS:** No RELAX, ocorre problema ao usar uma coluna intermediária (ex:  $R_3:y$ ). O sistema utiliza como se fosse a última coluna

## Resolva:

- Buscar o **código** dos médicos que têm/tiveram consultas com todos os pacientes
- Buscar o **CPF, nome e idade** dos pacientes que têm/tiveram consultas marcadas com todos os médicos
- Buscar o **nome** e o **CPF** dos pacientes que têm consultas marcadas com todos os médicos que atendem nos ambulatórios do **segundo andar**
- Verificar se todos os médicos ‘Ortopedistas’ atendem no mesmo ambulatório e, em caso positivo, mostrar o **número** e o **andar** deste ambulatório.

# Atualização de Relações

---

- Exclusão

- Notação:  $relação \leftarrow=== relação - expressãoConsulta$   
 $relação \leftarrow=== expressãoConsulta$

- Inclusão

- Notação:  $relação \leftarrow=== relação \cup Expr$   
onde  $Expr$  é um conjunto de tuplas

- Alteração

- Notação:  $\delta_{\{nome\_atributo \leftarrow=== Expr\}} (relação)$   
*(delta) onde  $Expr$  é uma expressão aritmética ou um valor constante*

# Atualização de Relações

---

Exemplos:

$R_1$

x	y	z
1	1	1
2	1	3

$R_2$

w	t	v
1	3	1
2	2	2
3	2	3

$$a) R_1 = R_1 - (\sigma_{x=1} R_1)$$

$$b) R_2 = (\sigma_{t=2} R_2)$$

$$c) R_1 = R_1 \cup \{(1,2,2), (1,2,3)\}$$

$$d) \text{temp} = \pi_w (\sigma_{t=2} R_2)$$

$$R_1 = R_1 \cup (\text{temp} \times \{(3,3)\})$$

$$e) \delta_{x < \dots x+1} R_1$$

$$f) \text{temp} = \sigma_{t=2} R_2$$

$$R_2 = R_2 - \text{temp}$$

$$\delta_{w < \dots w-1} \text{temp}$$

$$R_2 = R_2 \cup \text{temp}$$

# Atualização de Relações

---

## Resolva:

- Incluir registro de consulta para a paciente “Maria Goncalves Silva” no dia 02/03/2017 as 09:45, com a médica “Fernanda Jaicobson” e diagnóstico “gripe”
- O médico de código 100985 precisou transferir as suas consultas do dia 30/08/2016 para o dia 03/09/2016
- Remover os registros de consultas que não possuem uma doença associada, ou seja, doença = ‘-’