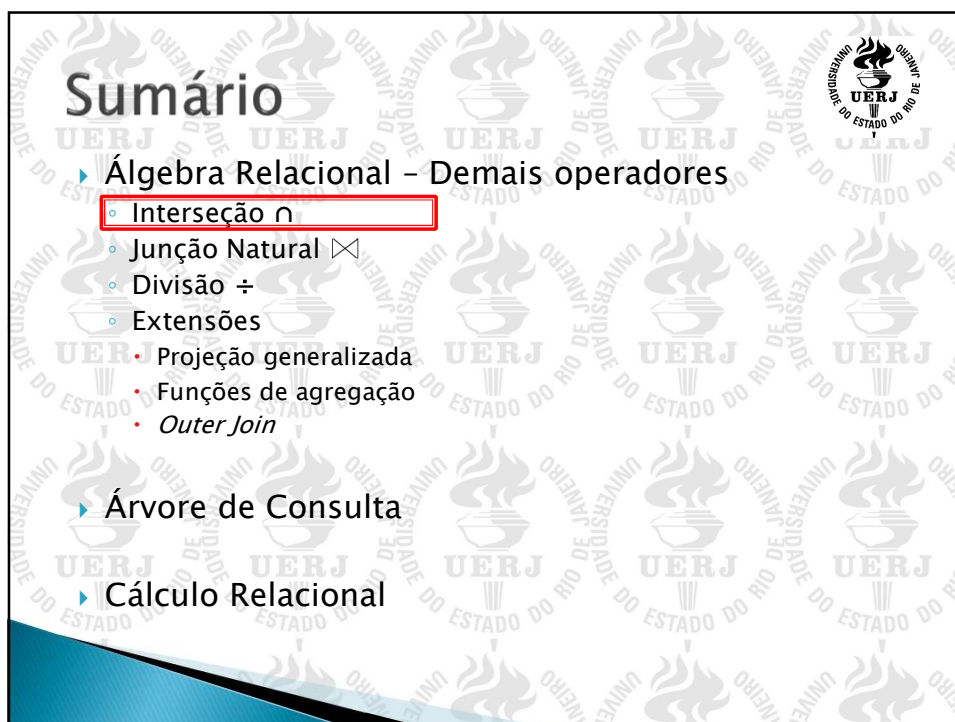


1



2

Operação INTERSECTION

▶ Notação: $R \cap S$
 ▶ Definida como:

$$r \cap s = \{ t \mid t \in r \text{ and } t \in s \}$$

 ▶ Assumindo que:

- R, S possuem a mesma aridade (mesmo número de atributos)
- Atributos de R e S são compatíveis (mesmo tipo)

 ▶ Note que:

$$r \cap s = r - (r - s)$$

 ▶ O resultado desta operação é uma relação que inclui todas as tuplas que existam tanto em R quanto em S

3

Operação INTERSECTION - Exemplo

▶ Interseção ($R \cap S$)

- Produz uma tabela resultado da **interseção** das tabelas operadas.

<u>Estudante</u>	
PrimeiroNome	ÚltimoNome
Maria	Silva
João	Souza
Pedro	Carneiro
Sandro	Vasquez
Aline	Brandão
Viviane	Gomes

<u>Instrutor</u>	
PrimeiroNome	ÚltimoNome
Aline	Brandão
Tatiana	Freitas
João	Souza
Ricardo	Marins

➡

Estudante \cap Instrutor	
PrimeiroNome	ÚltimoNome
João	Souza
Aline	Brandão

4

Operação INTERSECTION – Exemplo

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cap s$

A	B
α	2

5

Considerações – Teoria dos Conjuntos

- ▶ Ambas operações (união e interseção) são comutativas

$R \cup S = S \cup R, \text{ and } R \cap S = S \cap R$
- ▶ Ambas podem ser tratadas como operações n-árias aplicáveis para qualquer número de relações, sendo associativas:

$R \cup (S \cup T) = (R \cup S) \cup T, \text{ and } (R \cap S) \cap T = R \cap (S \cap T)$
- ▶ A operação diferença não é comutativa:

$R - S \neq S - R$

6

Sumário

- ▶ Álgebra Relacional – Demais operadores
 - Interseção \cap
 - Junção Natural \bowtie
 - Divisão \div
 - Extensões
 - Projeção generalizada
 - Funções de agregação
 - Outer Join
- ▶ Árvore de Consulta
- ▶ Cálculo Relacional

7

Operação JOIN

- ▶ Operação relacional binária
 - ▶ Junção natural
 - Sequência de um produto cartesiano seguido por uma seleção para identificar e selecionar as tuplas relacionadas de duas relações
 - ▶ Operação importante para qualquer banco de dados relacional com mais de uma relação porque permite processar relacionamentos entre relações
 - ▶ De uma forma geral:

$R \bowtie_{\langle \text{join condition} \rangle} S$
 - ▶ Onde,
 - R e S podem ser quaisquer relações que resultem de expressões da álgebra relacional.

8

Operação JOIN

Operação EQUIJOIN

- O uso mais comum de junção envolve condições de junção somente com comparação de igualdade.
- A operação de EQUIJOIN é a junção onde só é usado o operador de comparação =
- No resultado de um EQUIJOIN sempre existe um ou mais pares de atributos (cujos nomes não são necessariamente idênticos) que tenham valores idênticos em cada tupla.

Operação NATURAL JOIN

- Devido ao fato de que um de cada par de atributos com valores idênticos ser supérfluo, uma nova operação foi criada, chamada de junção natural - denotada por *
- A definição padrão de junção natural requer que dois atributos de junção ou cada par dos atributos de junção correspondentes tenham o **mesmo nome** em ambas relações.
- Se não for o caso, uma operação de "renomear" é aplicada primeiro.

9

Operação NATURAL-JOIN

- Dado que r e s sejam relações nos esquemas R e S , respectivamente.
- Então, $r \bowtie s$ é uma relação no esquema $R \cup S$ obtida da seguinte forma:
 - Considere cada par de tuplas t_r de r e t_s de s .
 - Se t_r e t_s tem o mesmo valor em cada atributo em $R \cap S$, adicione uma tupla t para o resultado, onde:
 - t tem o mesmo valor que t_r em r
 - t tem o mesmo valor que t_s em s
 - Exemplo:

$R = (A1, A2, A3, A4)$ e $S = (B1, A2, A4)$
 - Result schema = $(A1, A2, A3, A4, B1)$
 - $r \bowtie s$ is defined as:

$$\Pi_{r.A1, r.A2, r.A3, r.A4, s.B1} (\sigma_{r.A2 = s.A2 \wedge r.A4 = s.A4} (r \times s))$$

10

Operação JOIN – Exemplo


▶ Junção: $A \bowtie_{a=b} B$
 • Gera uma tabela que é a combinação das tabelas operadas segundo critérios impostos sobre atributos de uma e outra tabela.

Instrutor

PrimeiroNome	ÚltimoNome	Sexo	DataNascimento	Salário	CódDepto
Aline	Brandão	F	27/02/1987	4500	01
Tatiana	Freitas	F	11/03/1985	3700	02
João	Souza	M	08/01/1982	3200	01
Ricardo	Marins	M	10/10/1986	3200	02

Departamento

DCódDepto	DNome	DMatrGerente	DInicioGerente
01	Psicologia	12345	01/01/2011
02	Administração	67890	05/05/2010



11

Operação JOIN – Exemplo

Instrutor $\bowtie_{CódDepto=DCódDepto}$ Departamento

PrimeiroNome	ÚltimoNome	Sexo	DataNascimento	Salário	CódDepto	DNome	DMatrGerente	DInicioGerente
Aline	Brandão	F	27/02/1987	4500	01	Psicologia	12345	01/01/2011
Tatiana	Freitas	F	11/03/1985	3700	02	Administração	67890	05/05/2010
João	Souza	M	08/01/1982	3200	01	Psicologia	12345	01/01/2011
Ricardo	Marins	M	10/10/1986	3200	02	Administração	67890	05/05/2010

Para acompanhar...

DCódDepto	DNome	DMatrGerente	DInicioGerente
01	Psicologia	12345	01/01/2011
02	Administração	67890	05/05/2010

12

Operação JOIN – Exemplo

Relação r

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

V
F
F
V
V

Relação s

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

$r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

13

Sumário

- ▶ Álgebra Relacional – Demais operadores
 - Interseção \cap
 - **Junção Natural \bowtie**
 - Divisão \div
 - Extensões
 - Projeção generalizada
 - Funções de agregação
 - Outer Join
- ▶ Árvore de Consulta
- ▶ Cálculo Relacional

14

Operação DIVISION

- ▶ Operação relacional binária
 - A operação de divisão é aplicada a duas relações
 - $R(Z) \div S(X)$, onde X (*divisor*) é um subconjunto de Z (*dividendo*)
- ▶ Temos que $Y = Z - X$
 - $Z = X \cup Y$
 - Y : conjunto de atributos de R que não são atributos de S
- ▶ O resultado da divisão é uma relação $T(Y)$ que inclui uma tupla t se a tupla t_R aparece em R com $t_R[Y] = t$ e com $t_R[X] = t_S$ **para cada tupla** t_S em S
- ▶ Para uma tupla t aparecer no resultado T da divisão, os valores em t precisam aparecer em R na combinação com cada tupla de S

- ✓ Divisor tem subconjunto de atributos do dividendo
- ✓ Resultado – Parte do esquema que não está no divisor

15

Operação DIVISION – Exemplo

- ▶ Divisão: $R1 \div R2$
 - Opera duas tabelas, criando uma terceira com os atributos da primeira tabela que estejam associados a TODOS os atributos que existam na segunda tabela.

Relação R1

A	B
A01	B01
A02	B01
A03	B01
A01	B02
A03	B03
A01	B04
A02	B04
A03	B04

Relação R2

A
A01
A02
A03



$R1 \div R2$

B
B01
B04

16

Operação DIVISION



- ▶ Adequado para consultas que incluem a frase “para todo”
- ▶ Dado que r e s sejam relações nos esquemas R e S , respectivamente onde,
 - $R = (A_1, \dots, A_m, B_1, \dots, B_n)$
 - $S = (B_1, \dots, B_n)$
 - O resultado de $r \div s$ é uma relação no esquema

$$R - S = (A_1, \dots, A_m)$$

$$r \div s = \{ t \mid t \in \Pi_{R-S}(r) \wedge \forall u \in s (tu \in r) \}$$

- Onde tu significa a concatenação de tuplas t e u para produzir uma única tupla

17

Operação DIVISION



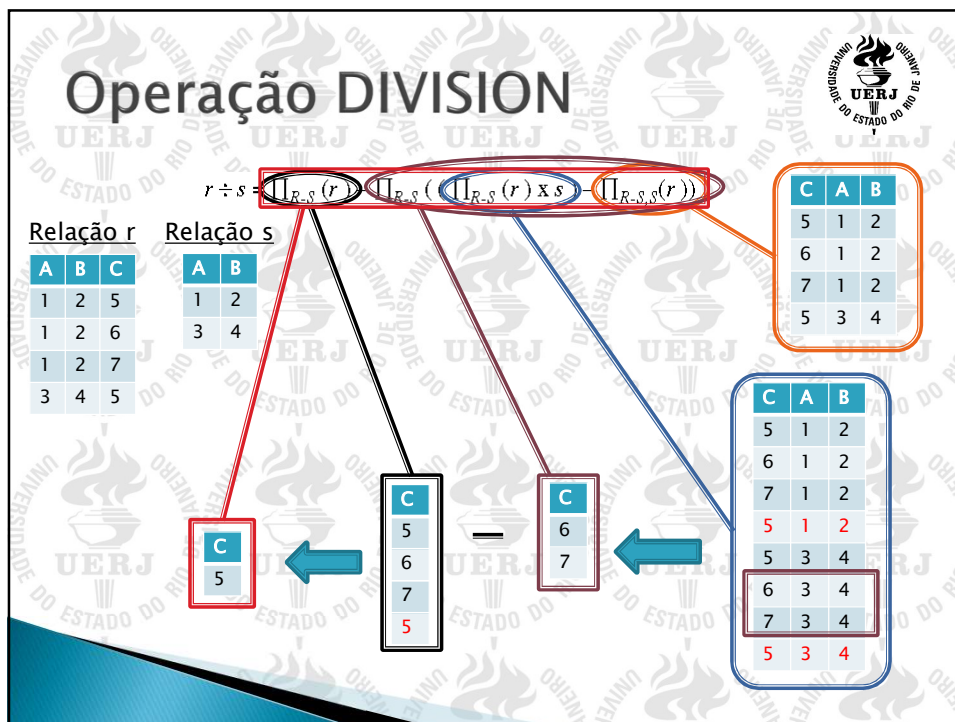
- ▶ Propriedade:
 - Dado que $q = r \div s$
 - Então: q é a maior relação que satisfaz $q \times s \subseteq r$
- ▶ Definição:
 - Dado que $r(R)$ e $s(S)$ sejam relações
 - $S \subseteq R$

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

Simplesmente reordena os atributos de r


Traz as tuplas t em $\Pi_{R-S}(r)$ tal que alguma tupla $u \in s$, $tu \notin r$.
Obs.: traz todas as combinações entre as relações participantes e que não existem em r .

18



19

Operação DIVISION



► **Resumo ($R \div S$):**

- Atributos (S) \subseteq Atributos (R)
- Relação resultante T:
 - Atributos (T) = { atributos (R) - atributos (S) }
 - Cada tupla de t, os valores de t devem aparecer em R combinados com cada tupla de S.
- Normalmente aplica-se a consultas que incluem “para todos”.

■ Relations r, s:

A	B	C	D	E
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

D	E
a	1
b	1

■ $r \div s$:

A	B	C
α	a	γ
γ	a	γ

20

Operação DIVISION – Exemplo

- ▶ Obtenha o nome dos empregados que trabalham em todos os projetos que Silva trabalha.

- $Silva \leftarrow \sigma_{nome = 'silva'}(Empregados)$
- $ProjSilva \leftarrow \pi_{codProj} (Alocação \bowtie_{matrEmp = matr} Silva)$
- $ProjEmp \leftarrow \pi_{codProj, matrEmp} (Alocação)$
- $TrabProjSilva \leftarrow (ProjEmp \div ProjSilva)$
- $Result \leftarrow \pi_{nome} (TrabProjSilva \bowtie_{matrEmp = matr} Empregado)$

21

Sumário

- ▶ Álgebra Relacional – Demais operadores

- Interseção \cap
- Junção Natural \bowtie
- Divisão \div
- Extensões
 - Projeção generalizada
 - Funções de agregação
 - Outer Join

- ▶ Árvore de Consulta

- ▶ Cálculo Relacional

22

Projeção Generalizada

- ▶ Estende a operação de projeção:
 - Permite funções aritméticas na lista de projeções

$$\Pi_{F_1, F_2, \dots, F_n}(E)$$

- ▶ E é qualquer expressão da álgebra relacional
- ▶ Cada F_1, F_2, \dots, F_n é uma expressão aritmética envolvendo constantes e atributos no esquema de E .

23

Operações Relacionais Adicionais

- ▶ Funções de agregações e agrupamentos
 - Um tipo de requisição que não pode ser expressada na álgebra relacional básica é especificada em funções de agregações matemáticas sobre coleções de valores do banco de dados.
 - Exemplos de tais funções incluem a recuperação da média ou salário total de todos os empregados ou número total de tuplas da relação empregado. Essas funções são usadas em consultas simples estatísticas que sumarizam informações das tuplas do banco de dados.
 - Funções comuns que são aplicadas nas coleções de valores numéricos incluem: SUM, AVERAGE, MAXIMUM e MINIMUM. A função COUNT é usada para contar tuplas ou valores.

24

Funções e Operações – Agregações

Função de agregação

- Pega uma coleção de valores e retorna um único valor como um resultado

avg: valor médio
min: valor mínimo
max: valor máximo

sum: soma de valores
count: número de valores

Operação de agregação na álgebra relacional

$G_1, G_2, \dots, G_n \text{ } \mathcal{G} F_1(A_1), F_2(A_2), \dots, F_m(A_m)(E)$

- E é qualquer expressão da álgebra relacional
- G_1, G_2, \dots, G_n é uma lista de atributos no qual agrupar (pode ser vazio)
- Cada F_i é uma função de agregação
- Cada A_i é um nome de atributo

25

Agregações – Exemplo

Relação r :

A	B	CC	CP
α	α	7	2
α	β	7	6
β	β	3	3
β	β	10	3

$\mathcal{G}_{\text{sum}(\text{CC})}(r)$

sum(CC)

27

$\Pi_{B, (CC-CP)}(r)$

B	CC - CP
α	5
β	1
β	0
β	7

26

Agregações – Exemplo

► Esquema “Banco”

- agência (nome_agencia, cidade_agencia, ativos)
- cliente (nome_cliente, endereço_cliente, cidade_cliente)
- conta (numero_conta, nome_agencia, saldo)
- emprestimo (codigo_emprestimo, nome_agencia, valor)
- depositante (nome_cliente, numero_conta)
- tomador_emprestimo (nome_cliente, codigo_emprestimo)

27

Agregações – Exemplo

Relação *conta* agrupada por *nome_agencia*:

nome_agencia	numero_conta	saldo
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

nome_agencia \mathcal{G} sum(saldo) (conta)

nome_agencia	sum(saldo)
Perryridge	1300
Brighton	1500
Redwood	700

28

Agregações

- ▶ O resultado da agregação não tem um nome
 - Pode usar a operação de renomear para atribuir um nome
 - Por conveniência, é permitido a renomeação como parte da operação agregada

nome_agencia *g* sum(saldo) as soma_saldo (conta)



29

Valores nulos

- ▶ É possível que as tuplas tenham valores nulos (*null*) para alguns de seus atributos
- ▶ *Null* significa um valor desconhecido ou um valor que não existe
- ▶ O resultado de qualquer expressão aritmética envolvendo *null* é *null*.
- ▶ Funções de agregações simplesmente ignoram valores *null* (como no SQL)
- ▶ Para eliminar duplicatas e agrupamentos, *null* é tratado como qualquer outro valor
 - E dois *nulls* são assumidos como sendo o mesmo (como em SQL)

30

Valores nulos

- ▶ Comparações com valores *null* retornam um valor verdadeiro especial: *desconhecido*
 - Se *falso* for usado ao invés de *desconhecido*, então $\text{not}(A < 5)$ não seria equivalente a $A \geq 5$
- ▶ Operadores lógicos usando o valor verdadeiro *desconhecido*:
 - OR:
 - $(\text{desconhecido or verdadeiro}) = \text{verdadeiro}$
 - $(\text{desconhecido or falso}) = \text{desconhecido}$
 - $(\text{desconhecido or desconhecido}) = \text{desconhecido}$
 - AND:
 - $(\text{verdadeiro and desconhecido}) = \text{desconhecido}$
 - $(\text{falso and desconhecido}) = \text{falso}$
 - $(\text{desconhecido and desconhecido}) = \text{desconhecido}$
 - NOT:
 - $(\text{not desconhecido}) = \text{desconhecido}$
- ▶ Resultado do predicado de seleção é tratado como *falso* se ele avalia o *desconhecido*.

31

Operação OUTER JOIN

- ▶ Na junção natural, as tuplas que não possuem relações com as tuplas da outra relação são eliminadas do resultado da junção. As tuplas com valores nulos (*null*) nos atributos de junção também são eliminadas.
 - Esses valores levam a perda de informação
- ▶ Um conjunto de operações, chamado de *outer joins*, podem ser usados quando nós precisamos manter todas as tuplas em R ou todas em S ou todas de ambas relações no resultado da junção, ao invés de somente as tuplas que se relacionam com a outra relação.

32

Operação OUTER JOIN

Tipos de OUTER JOIN

- A operação LEFT OUTER JOIN mantém cada tupla da primeira relação (ou da esquerda) R em $R \bowtie S$; se nenhuma tupla é encontrada relacionada em S, então os atributos de S no resultado da junção são preenchidos com valores nulos (*null*).
- Uma operação similar, RIGHT OUTER JOIN, mantém cada tupla na segunda relação (ou da direita) S no resultado de $R \bowtie S$.
- Uma terceira operação, FULL OUTER JOIN, denotada por \bowtie , mantém todas as tuplas em ambas relações (esquerda e direita) quando nenhuma tupla relacionada é encontrada, preenchendo elas com valores nulos quando necessário.

33

OUTER JOIN – Exemplo

Relação *empréstimo*:

codigo_empréstimo	nome_agencia	valor
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

Relação *tomador_empréstimo*:

nome_cliente	codigo_empréstimo
Jones	L-170
Smith	L-230
Hayes	L-155

34

OUTER JOIN – Exemplo

► Primeiro: lembrando o inner join (natural join)

◦ `emprestimo` ⋈ `tomador_emprestimo`

<code>codigo_emprestimo</code>	<code>nome_agencia</code>	<code>valor</code>	<code>nome_cliente</code>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

35

OUTER JOIN – Exemplo

► Left Outer Join

`emprestimo` ⋈ `tomador_emprestimo`

<code>codigo_emprestimo</code>	<code>nome_agencia</code>	<code>valor</code>	<code>nome_cliente</code>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>

► Right Outer Join

`emprestimo` ⋈ `tomador_emprestimo`

<code>codigo_emprestimo</code>	<code>nome_agencia</code>	<code>valor</code>	<code>nome_cliente</code>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	<i>null</i>	<i>null</i>	Hayes

36

OUTER JOIN – Exemplo

▶ Full Outer Join

emprestimo \bowtie tomador_emprestimo

codigo_emprestimo	nome_agencia	valor	nome_cliente
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>
L-155	<i>null</i>	<i>null</i>	Hayes

37

Sumário

- ▶ Álgebra Relacional – Demais operadores
 - Interseção \cap
 - Junção Natural \bowtie
 - Divisão \div
 - Extensões
 - Projeção generalizada
 - Funções de agregação
 - Outer Join
- ▶ Árvore de Consulta
- ▶ Cálculo Relacional

38

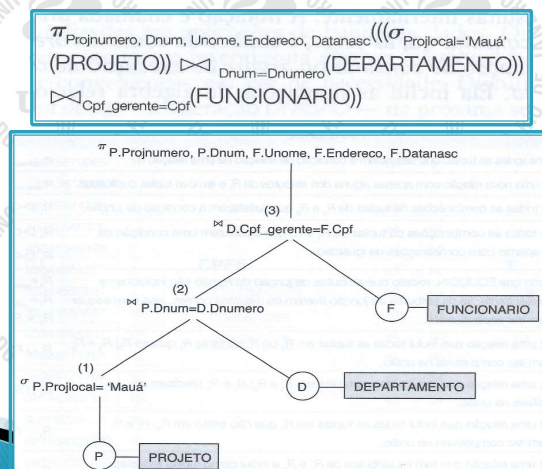
Árvore de consulta

- ▶ Estrutura de dados usada para representar as consultas internamente.
- ▶ Relações de entrada
 - Nós folhas
- ▶ Operações da álgebra relacional
 - Nós internos
- ▶ Execução
 - Executar uma operação de nó interno sempre que seus operandos (nós filhos) estiverem disponíveis
 - Substituir esse nó interno pela relação que resulta da execução da operação
 - Execução termina quando o nó raiz é executado e produz a relação de resultado para a consulta


39

Árvore de consulta – Exemplo

- ▶ Para cada projeto localizado em 'Mauá', liste o número dele, número do seu departamento, último nome, endereço e data de nascimento do gerente do departamento.



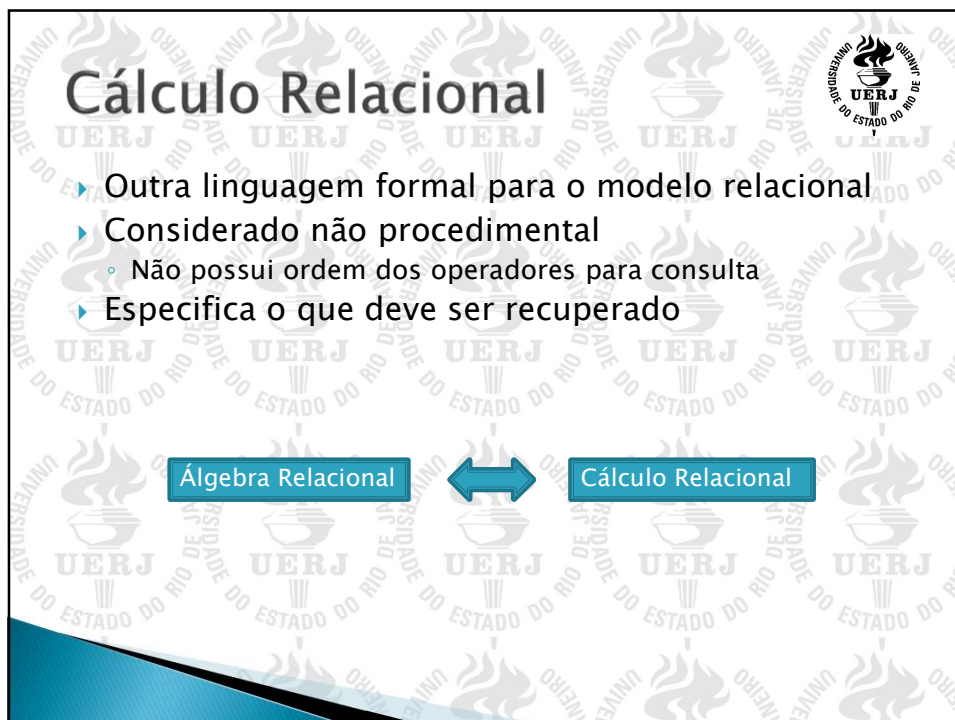
40




Sumário

- ▶ Álgebra Relacional – Demais operadores
 - Interseção \cap
 - Junção Natural \bowtie
 - Divisão \div
 - Extensões
 - Projeção generalizada
 - Funções de agregação
 - *Outer Join*
- ▶ **Árvore de Consulta**
- ▶ Cálculo Relacional

41



42



Cálculo Relacional

- ▶ Cálculo relacional de tupla
 - Variáveis percorrendo tuplas
 - Operação de seleção (álgebra relacional)
 - Ex.: $\{t \mid \text{Funcionario}(t) \text{ AND } t.\text{sexo} = 'M' \text{ AND } t.\text{data_de_nascimento} < '31/12/1950'\}$
- ▶ Cálculo de domínio
 - Variáveis percorrem valores isolados dos domínios de atributos
 - Uma variável para cada atributo, mesmo que não participe das condições
 - $\{u \mid (n) (d) (s) (m) (\text{Funcionario}(ndsm) \text{ AND } d < '31/12/1950' \text{ AND } s = 'M')\}$

N: nome
d: data de nascimento
S: sexo
M: matrícula
U: nome (atributo de projeção)

43



Esquema Relacional

- ▶ EMPREGADO (Ident, Nome, Sal, End, Sexo, DataNasc, DepNum, SuperIdent)
- ▶ DEPARTAMENTO (Num, Nome, IdentGer)
- ▶ PROJETO (Num, Nome, Local, DepNum)
- ▶ TRABALHA_NO (IdentEmp, ProjNum, HRS)
- ▶ DEPENDENTE (Nome, Sexo, DataNasc, Parentesco, IdentEmp)
- ▶ DEPLOC (DepNum, Local)

44

Exercício 1

- Listar apenas os projetos (número e nome) que são localizados em um dos locais onde o departamento, que controla este projeto, está localizado.

```
δ Local→LocalDep (DepLoc)
T1 ← Projeto ⋈ DepLoc
Resposta ← Π(Num, Nome) (σ(Local = LocalDep) (T1))
```

```
T1 ← Projeto ⋈ DepLoc
Resposta ← Π(Num, Nome) (T1)
```

45

Exercício 2

- Quais as identidades e nomes dos empregados, além dos números de departamento dos quais eles são gerentes, apenas se o forem?

```
(Left ou Full Outer Join)
EmpGer ← Empregado ⋈ Departamento
Resposta ← Π(Ident, Empregado, Nome, Num) (EmpGer)
```

46

Exercício 3



- Listar o nome dos empregados que trabalham em todos os projetos em que o empregado de nome Manuel Silva trabalha (supor que não existam dois empregados com este mesmo nome).

```
Manuel ← σ(Nome = 'Manuel Silva') (Empregado)

Proj_Manuel ← π(ProjNum) (Trabalha_no ⋈(IdentEmp = Ident) Manuel)

IdentProjs ← π(ProjNum, IdentEmp) (Trabalha_no)

Idents ← IdentProjs ÷ Proj_Manuel

Resposta ← π(Nome) (Idents ⋈(IdentEmp = Ident) Empregado)
```

Obs. Se houver mais de um Manuel, incluir Identidade na resposta. Por sinal, na resposta também estará o(s) Manuel(éis). Para retirá-lo(s) estender com diferença relacional ou usar ≠

47

Exercício 4



- Quais empregados (identidade e nome) que estão trabalhando menos de 40 horas por semana, considerando-se todos os projetos em que trabalham?

```
Emps_HrsSem ← Ident ⋈Sum(HRS) (Trabalha_no)

Idents_Menos40 ← σ(Sum(HRS) < 40) (Emps_HrsSem)

Resposta ← π(Ident, Nome) (Idents_Menos40 ⋈(IdentEmp = Ident) Empregado)
```

48

Exercício 5

- Em qual (quais) departamento(s) trabalha(m) o(s) empregado(s) que recebem o maior salário pago pela empresa?

$\text{MaiorSal} \leftarrow \mathcal{G}_{\text{Max}(\text{Sal})}(\text{Empregado}) \text{ as MSal}$

$\text{Resposta} \leftarrow \Pi_{(\text{Ident}, \text{Nome}, \text{DepNum})}(\text{Empregado} \bowtie_{(\text{Sal} = \text{MSal})} \text{MaiorSal})$

49

Exercício 6

- Quais são os dependentes (nome, parentesco e data de nascimento) que são mais velhos do que o empregado do qual dependem?

$\delta \text{ Nome} \rightarrow \text{NomeDep} \quad (\text{Dependente})$

$\delta \text{ Sexo} \rightarrow \text{SexoDep} \quad (\text{Dependente})$

$\delta \text{ DataNasc} \rightarrow \text{NascDep} \quad (\text{Dependente})$

$T1 \leftarrow \sigma_{(\text{NascDep} < \text{DataNasc})}(\text{Dependente} \bowtie_{(\text{Ident} = \text{IdentEmp})} \text{Empregado})$

$\text{Resposta} \leftarrow \Pi_{(\text{Nome}, \text{Parentesco}, \text{NascDep})}(T1)$

50