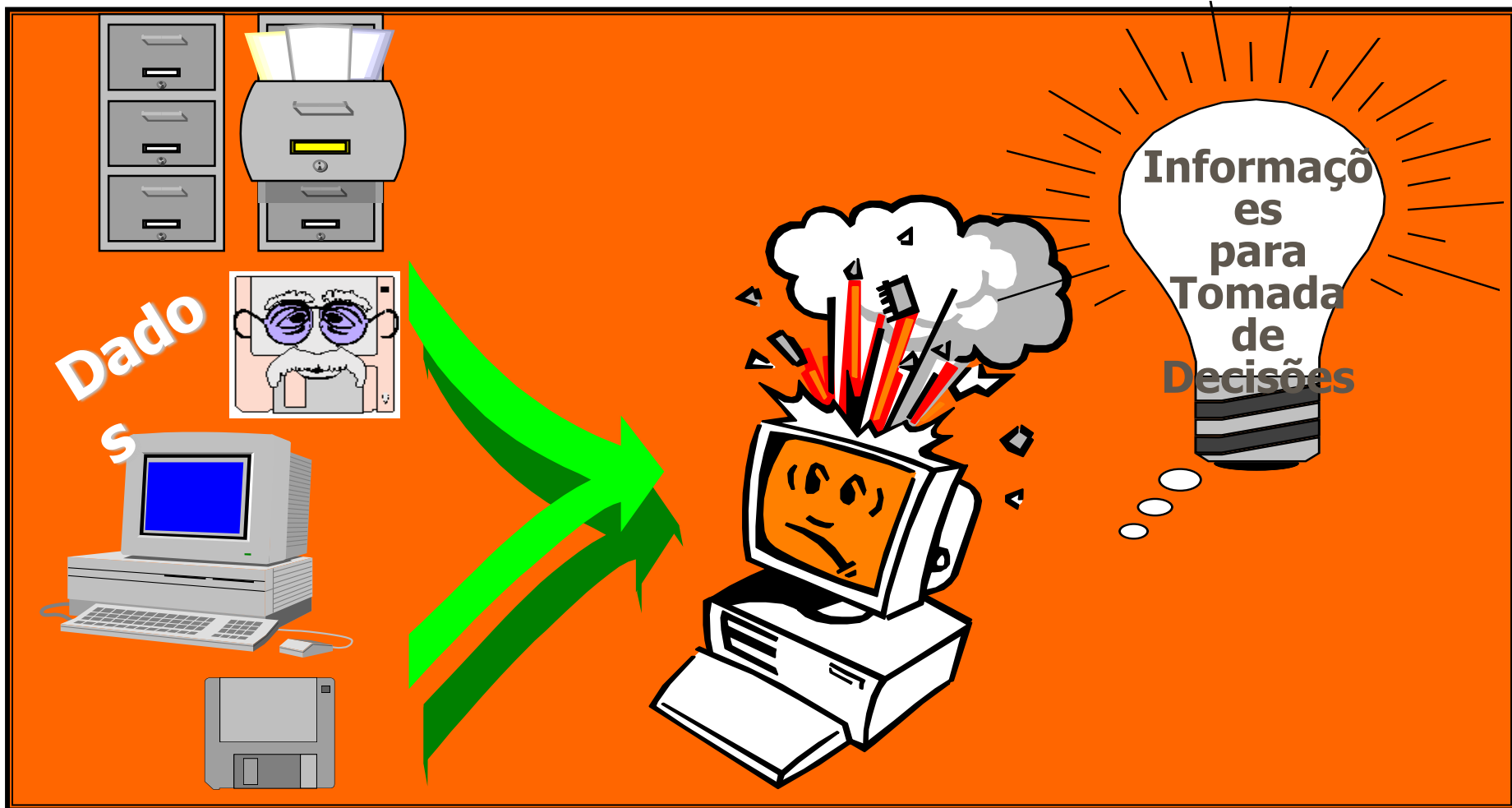


Bancos de Dados



Deborah Ribeiro Carvalho
2019

Declarativa x Operacional

- Declarativa:
- Operacional:

Declarativa x Operacional

- Declarativa:

- Quero um misto-quente

- Operacional:

- Quero duas fatias de pão de forma, recheadas com uma fatia de queijo e uma fatia de presunto. Tudo isto bem tostado.

▶ Básicas

- ▶ Seleção (σ) - seleciona um subconjunto de linhas de uma relação
- ▶ Projeção (π) – apaga colunas desnecessárias de uma relação
- ▶ Produto cartesiano (\times) – permite combinar duas relações
- ▶ União (\cup) - tuplas na relação 1 e na relação 2
- ▶ Diferença ($-$) – tuplas na relação 1 mas não na relação 2
- ▶ Renomeação (ρ) – renomeia tabela
- ▶ Atribuição (\leftarrow) – Atribui valores a variáveis

▶ Derivadas

- ▶ Junção, interseção, divisão

- Seleciona tuplas da relação argumento que satisfaçam à condição de seleção

$\sigma_{\text{condição_seleção}}$ (relação argumento)

- pode envolver operadores de comparação
(=, <, ≤, >, ≥, ≠)
- pode combinar condições usando-se \wedge , \vee , \neg

- relação
- resultado de alguma operação da álgebra relacional

- Produz uma nova relação contendo um subconjunto vertical da relação argumento, sem duplicações

$\pi_{\text{lista_atributos}}$ (relação argumento)

- lista de atributos
- os atributos são separados por vírgula

- relação
- resultado de alguma operação da álgebra relacional

Álgebra Relacional

Seleção e Projeção - exercício de fixação

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Quais são os dados dos médicos com idade superior a 30 anos?

σ idade > 30 (medicos)

Álgebra Relacional

Seleção e Projeção - exercício de fixação

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome e CPF dos médicos com superior a 30 anos?

Π nome, CPF (σ idade > 30 (medicos))

- associa uma relação argumento a uma relação temporária
- permite o uso da relação temporária em expressões subsequentes

relação temporária \leftarrow relação argumento

• resultado de alguma operação da álgebra relacional

• relação

- Liste o número e o nome de todos os clientes que possuam saldo inferior a R\$ 200,00 e que morem na Rua X.

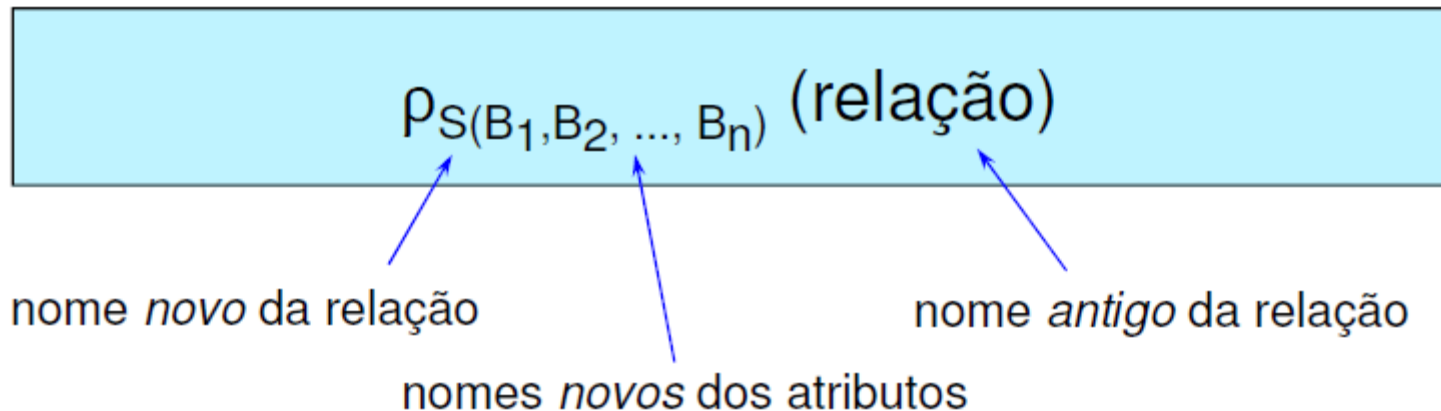
$$\pi_{\text{nro_cli}, \text{nome_cli}} (\sigma_{\text{saldo_dev} < 200,00 \wedge \text{end_cli} = \text{"Rua X"}} (\text{cliente}))$$

- Usando atribuição
 - $\text{temp} \leftarrow \sigma_{\text{saldo_dev} < 200,00 \wedge \text{end_cli} = \text{"Rua X"}} (\text{cliente})$
 - $\pi_{\text{nro_cli}, \text{nome_cli}} (\text{temp})$

- Características adicionais
 - permite renomear os atributos de relações intermediárias e final
 - $R(\text{código}, \text{nome}) \leftarrow \pi_{\text{nro_cli}, \text{nome_cli}}(\text{temp})$
- Observações
 - não adiciona potência adicional à álgebra relacional
 - geralmente utilizada para expressar consultas complexas

- ▶ Armazena o resultado de uma expressão algébrica em uma variável de relação
 - ▶ permite o processamento de uma consulta complexa em etapas
- ▶ Notação
 - ▶ $\text{nomeVariável} \leftarrow \text{expressãoÁlgebra}$
- ▶ Exemplo
 - ▶ $r1 \leftarrow \sigma_{\text{nome}='bob'}(\text{estudante})$

- Renomeia
 - nome da relação
 - nomes dos atributos da relação
 - nome da relação e nomes dos atributos



- ▶ **Notação:** $\rho_S(R)$ ou $\rho_{S(A_1, A_2, \dots)}(R)$
 - ▶ Renomeia R para S ou renomeia R para S com atributos renomeados A_1, A_2, \dots
- ▶ **Entrada:** Tabela (R)
- ▶ **Propósito:** redefinir nome tabelas / ou colunas num contexto
- ▶ **Saída:** Tabela renomeada com mesmas linhas de R
- ▶ Usada para
 - ▶ Útil para auto-relacionamentos, onde precisamos fazer a junção de uma tabela com ela mesma, e nesse caso cada versão da tabela precisa receber um nome diferente da outra.
 - ▶ Cria colunas idênticas numa junção natural

- Exemplos

- $\rho_{\text{comprador}}(\text{cliente})$
- $\rho_{(\text{código, nome, rua, saldo, vendedor})}(\text{cliente})$
- $\rho_{\text{comprador}(\text{código, nome, rua, saldo, vendedor})}(\text{cliente})$

- Observação

- indicada para ser utilizada quando uma relação é usada mais do que uma vez para responder à consulta

$DEP4_SAL2000 \leftarrow \sigma_{NumDep = 4 \text{ AND } Sal\acute{a}rio > 2000}(EMPREGADO)$

$RESULT \leftarrow \pi_{NumBI, NomeP, NomeF}(DEP4_SAL2000)$

DEP4_SAL2000	NomeP	NomeF	NumBI	...	Salário	NumDep
	João	Santos	798764544	...	2500	4
	Ana	Feio	342342324	...	3000	4

RESULT	NumBI	NomeP	NomeF
	798764544	João	Santos
	342342324	Ana	Feio

$\rho_{DEP4_SAL2000}(\sigma_{NumDep = 4 \text{ AND } Sal\acute{a}rio > 2000}(EMPREGADO))$
 $\rho_{RESULT}(BI, Nome, Apelido)(\pi_{NumBI, NomeP, NomeF}(DEP4_SAL2000))$

RESULT	BI	Nome	Apelido
	798764544	João	Santos
	342342324	Ana	Feio

- Unárias

- seleção
- projeção
- renomear

operam sobre uma
única relação

- Binárias

- produto cartesiano
- união
- diferença de conjuntos
- intersecção de conjuntos
- junção natural
- divisão

operam sobre duas
relações

- ▶ **Notação:** $R \times S$
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera combinações de linhas das duas tabelas
- ▶ **Saída:** Para cada linha r em R e cada linha s em S , gerar a tupla rs

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

vendedor (cod_vend, nome_vend)

cod_vend	nome_vend
1	Adriana
2	Roberto

Álgebra Relacional

Produto cartesiano

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

grau: número de atributos de cliente + número de atributos de vendedor

número de tuplas: número de tuplas de cliente * número de tuplas de vendedor

- Considere as seguintes relações
 - usuário (cliente_nome, gerente_nome)
 - cliente (cliente_nome, rua, cidade)

cliente_nome	gerente_nome
Márcia	Manoel
Rodrigo	Maria

cliente_nome	rua	cidade
Márcia	Rua X	Itambé
Rodrigo	Rua X	Maringá

- Liste o nome de todos os usuários atendidos pelo gerente Manoel, assim como as cidades nas quais eles vivem.

Álgebra Relacional

Produto cartesiano – exemplo de fixação

- usuário (cliente_nome, gerente_nome)
- cliente (cliente_nome, rua, cidade)

$\text{temp}_1 \leftarrow \pi_{\text{cliente_nome}} (\sigma_{\text{gerente_nome} = \text{"Manoel"}} (\text{usuário}))$

$\text{temp}_2 \leftarrow \text{temp}_1 \times \text{cliente}$

$\text{temp}_3 \leftarrow \sigma_{\text{temp1.cliente_nome} = \text{cliente.cliente_nome}} (\text{temp}_2)$

$\pi_{\text{temp1.cliente_nome}, \text{cidade}} (\text{temp}_3)$

- Considere a seguinte relação
– cliente (cliente_nome, rua, cidade)

cliente_nome	rua	cidade
Márcia	Rua X	Itambé
Rodrigo	Rua X	Maringá
Cristina	Rua XTZ	Maringá
Sofia	Rua X	Maringá
Ricardo	Rua AAA	Itambé

- Liste o nome dos clientes que moram na mesma rua e na mesma cidade que Rodrigo

– cliente (cliente_nome, rua, cidade)

- Primeiro passo

- determinar o nome da rua e o nome da cidade na qual Rodrigo mora

$\text{temp}_1 \leftarrow \pi_{\text{rua}, \text{cidade}} (\sigma_{\text{cliente_nome} = \text{"Rodrigo"}} (\text{cliente}))$

- relação resultado temp_1

rua	cidade
Rua X	Maringá

Álgebra Relacional

Produto cartesiano – exemplo de fixação

– cliente (cliente_nome, rua, cidade)

♦ Segundo passo

– realizar o produto cartesiano das relações

$$\text{temp}_2 \leftarrow \text{temp}_1 \times \text{cliente}$$

– relação resultado temp_2

$\text{temp}_1.\text{rua}$	$\text{temp}_1.\text{cidade}$	cliente_nome	cliente.rua	cliente.cidade
Rua X	Maringá	Márcia	Rua X	Itambé
Rua X	Maringá	Rodrigo	Rua X	Maringá
Rua X	Maringá	Cristina	Rua XTZ	Maringá
Rua X	Maringá	Sofia	Rua X	Maringá
Rua X	Maringá	Ricardo	Rua AAA	Itambé

– cliente (cliente_nome, rua, cidade)

♦ Terceiro passo

– eliminar informações indesejadas

$$\text{temp}_3 \leftarrow \sigma_{\text{cliente_nome} \neq \text{"Rodrigo"}}(\text{temp}_2)$$

– relação resultado temp₃

temp ₁ .rua	temp ₁ .cidade	cliente_nome	cliente.rua	cliente.cidade
Rua X	Maringá	Márcia	Rua X	Itambé
Rua X	Maringá	Cristina	Rua XTZ	Maringá
Rua X	Maringá	Sofia	Rua X	Maringá
Rua X	Maringá	Ricardo	Rua AAA	Itambé

– cliente (cliente_nome, rua, cidade)

♣ Quarto passo

– exibir as informações solicitadas

$$\pi_{\text{cliente_nome}} (\sigma_{\text{temp}_1.\text{rua} = \text{cliente.rua} \wedge \text{temp}_1.\text{cidade} = \text{cliente.cidade}} (\text{temp}_3))$$

– relação resultado

cliente_nome
Sofia

– cliente (cliente_nome, rua, cidade)

- Liste o nome dos clientes que moram na mesma rua e na mesma cidade que Rodrigo

$\text{temp}_1 \leftarrow \pi_{\text{rua}, \text{cidade}} (\sigma_{\text{cliente_nome} = \text{"Rodrigo"}} (\text{cliente}))$

$\text{temp}_2 \leftarrow \text{temp}_1 \times \text{cliente}$

$\text{temp}_3 \leftarrow \sigma_{\text{cliente_nome} \neq \text{"Rodrigo"}} (\text{temp}_2)$

$\pi_{\text{cliente_nome}} (\sigma_{\text{temp}_1.\text{rua} = \text{cliente.rua} \wedge \text{temp}_1.\text{cidade} = \text{cliente.cidade}} (\text{temp}_3))$

Álgebra Relacional

Produto cartesiano – exemplo de fixação

- Obtenha o nome dos dependentes dos empregados do sexo feminino.

$EMP_FEM \leftarrow \sigma_{Sexo = 'F'}(EMPREGADO)$

$FEM_DEPS \leftarrow \sigma_{NumBI = EmpBI}(EMP_FEM \times DEPENDENTES)$

$RESULT \leftarrow \pi_{Nome}(FEM_DEPS)$

Alternativa de resolução mais eficiente:

$EMP_FEM \leftarrow \pi_{NumBI}(\sigma_{Sexo = 'F'}(EMPREGADO))$

$DEPS_NOME \leftarrow \pi_{EmpBI, Nome}(DEPENDENTES)$

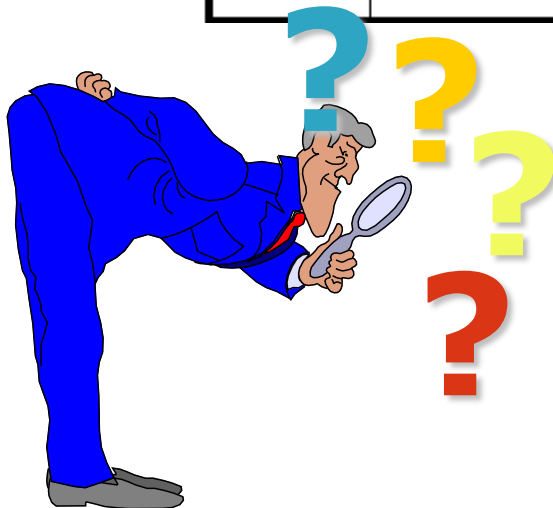
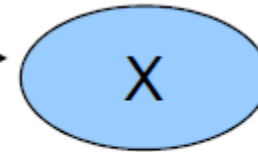
$FEM_DEPS \leftarrow \sigma_{NumBI = EmpBI}(EMP_FEM \times DEPS_NOME)$

$RESULT \leftarrow \pi_{Nome}(FEM_DEPS)$

- ▶ A ordem das colunas não é importante
 - ▶ Assim a operação é comutativa
 - ▶ $R \times S = S \times R$

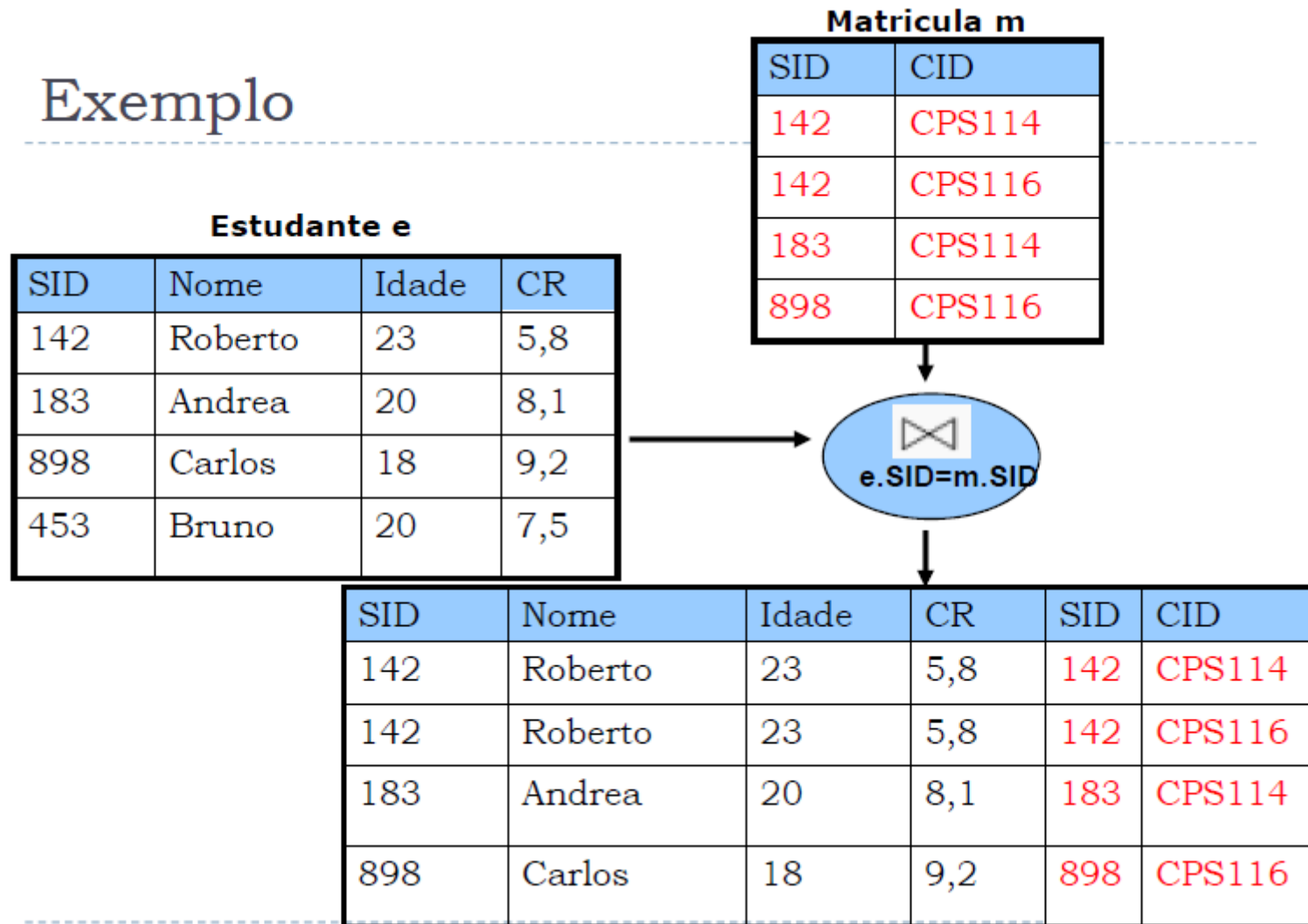
SID	Nome	Idade	CR
142	Roberto	23	5,8
183	Andrea	20	8,1
898	Carlos	18	9,2
453	Bruno	20	7,5

SID	CID
142	CPS116
183	CPS114
898	CPS116



SID	Nome	Idade	CR	SID	CID
142	Roberto	23	5,8	142	CPS116
142	Roberto	23	5,8	183	CPS114
142	Roberto	23	5,8	898	CPS116
183	Andrea	20	8,1	142	CPS116
.....

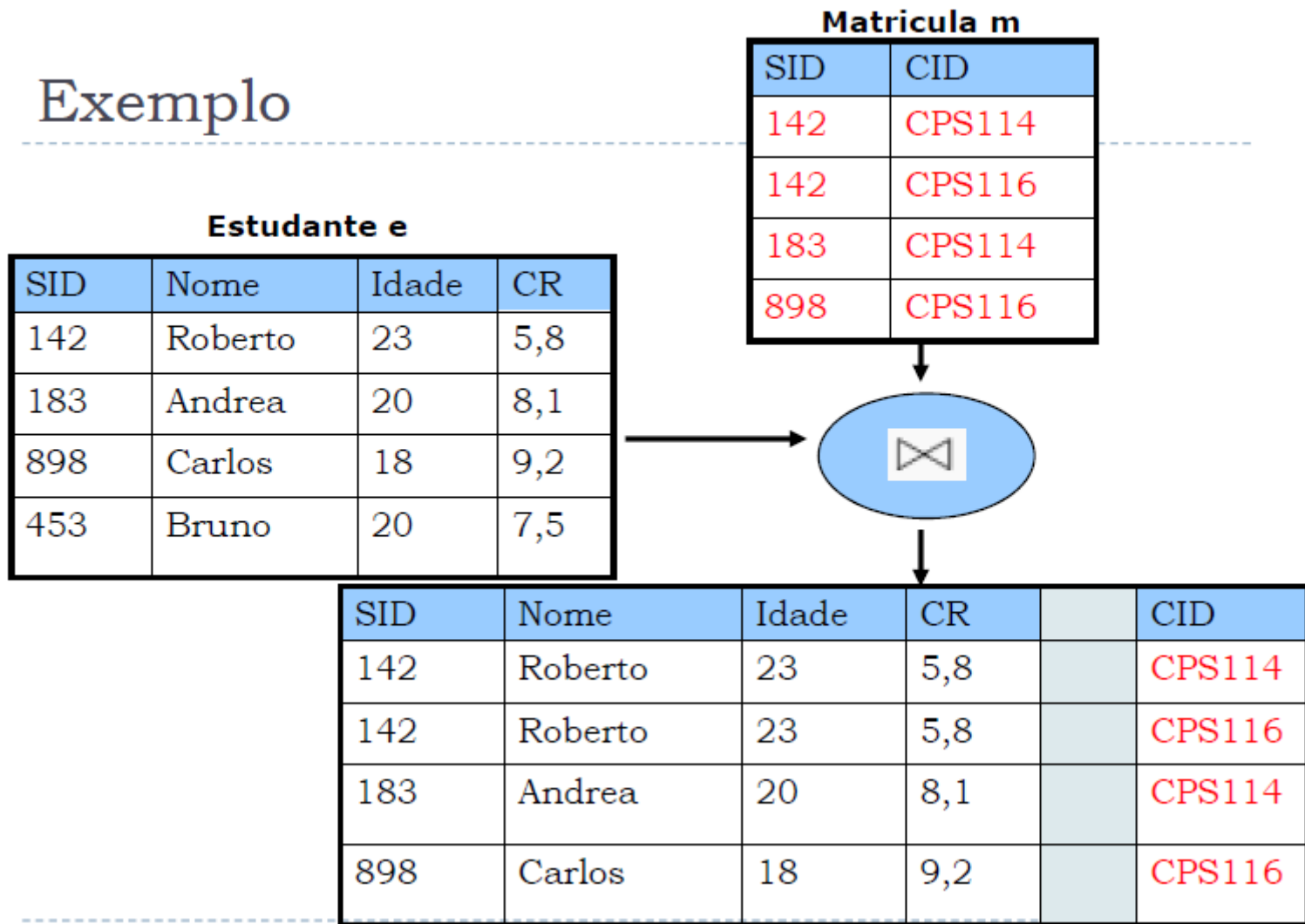
Exemplo



- ▶ **Notação:** $R \bowtie S$
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** relaciona linhas das tabelas
 - ▶ Reforça a igualdade de seus atributos
 - ▶ Elimina 1 cópia dos atributos comuns
- ▶ **Saída:** Para cada linha r em R e cada linha s em S , gerar a tupla rs se, e somente se, atenderem a condição p

- ▶ **Notação:** $R \bowtie_p S$
 - ▶ “p” é a condição/predicado da junção
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Para cada linha r em R e cada linha sem S, gerar a tupla rs se, e somente se, atenderem a condição p
- ▶ Abreviação para $\sigma_p (R \times S)$

Exemplo



- Obtenha o nome dos dependentes dos empregados do sexo feminino.

$$\text{EMP_FEM} \leftarrow \sigma_{\text{Sexo} = 'F'}(\text{EMPREGADO})$$

$$\text{RESULT} \leftarrow \pi_{\text{Nome}}(\text{EMP_FEM} \bowtie_{\text{NumBI} = \text{EmpBI}} \text{DEPENDENTES})$$

- Obtenha o nome e o endereço de todos os empregados que trabalham no departamento de Produção.

$$\text{EMP_DEP} \leftarrow \text{EMPREGADO} \bowtie_{\text{NumDep} = \text{Num}} \text{DEPARTAMENTO}$$

$$\text{RESULT} \leftarrow \pi_{\text{NomeP}, \text{NomeF}, \text{Endereço}}(\sigma_{\text{Nome} = 'Produção'}(\text{EMP_DEP}))$$

- Para todos os projectos localizados no Porto, obtenha o nome do projecto e o último nome do respectivo gerente.

$$\text{PROJ_PORTO} \leftarrow \sigma_{\text{PLocal} = 'Porto'}(\rho_{(\text{PNome}, \text{PNum}, \text{PLocal}, \text{PDep})}(\text{PROJECTO}))$$

$$\text{PORTO_DEP} \leftarrow \text{PROJ_PORTO} \bowtie_{\text{PDep} = \text{Num}} \text{DEPARTAMENTO}$$

$$\text{RESULT} \leftarrow \pi_{\text{PNome}, \text{NomeF}}(\text{PORTO_DEP} \bowtie_{\text{GerenteBI} = \text{NumBI}} \text{EMPREGADO})$$

- **Ex:** encontrar todos os nomes de clientes que tenham um empréstimo no banco, bem como o total emprestado.

Utilizando apenas as operações fundamentais, poderia ser resolvido assim:

$$\pi_{\text{nome_cliente, emprestimo.num_emprestimo, total}}(\sigma_{\text{devedor.num_emprestimo = emprestimo.num_emprestimo}}(\text{devedor} \times \text{emprestimo}))$$

No entanto, é possível simplificar esta operação utilizando o símbolo da junção natural: \bowtie
A junção natural forma um produto cartesiano das duas relações, e já executa uma seleção, fazendo a equivalência dos atributos que aparecem em ambos os esquemas da relação, e por último, remove os atributos em duplicidade.

$$\pi_{\text{nome_cliente, num_emprestimo total}}(\text{devedor} \bowtie \text{empréstimo})$$

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome do médico que atende no ambulatorio 15?

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome do médico que atende no ambulatorio 15?

$\Pi \text{ nome } (\sigma \text{ nroa} = 15 \text{ (medicos)})$

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome do médico que atende no ambulatorio 15, bem como o respectivo andar?

Ambulatórios(nroa, andar, capacidade)

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

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

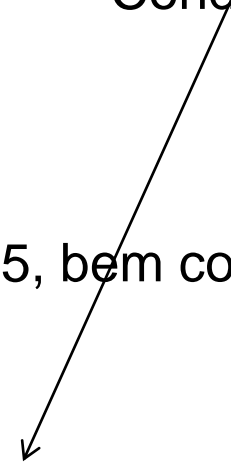
Consultas(codm, codp, data, hora)

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

Condicao

Qual é o nome do médico que atende no ambulatorio 15, bem como o respectivo andar?

Π medicos.nome, ambulatórios.andar
(σ medicos.nroa = 15 ((medicos) [X] (ambulatorios)))



Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome do médico que atende no ambulatorio 15, bem como o respectivo andar?

```
II medicos.nome, ambulatórios.andar  
  (σ medicos.nroa = 15 ((medicos) [X]  
    medicos.nroa = ambulatorios.nroa (ambulatorios)))
```

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome dos médicos que tiveram consulta em maio de 2019?

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome dos médicos que tiveram consulta em maio de 2019?

PASS01 \leftarrow (σ mes(consultas.data) = 5 \wedge
ano(consultas.data) = 2019 (consultas))

Π medicos.nome ((medicos) [X]
medicos.codm = PASS01.codm (PASS01))

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome dos médicos e respectivos pacientes que tiveram consulta em maio de 2019?

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome dos médicos e respectivos pacientes que tiveram consulta em maio de 2019?

PASS01 \leftarrow (σ mes(consultas.data) = 5 \wedge
ano(consultas.data) = 2019 (consultas))

PASS02 \leftarrow ((medicos) [X]
medicos.codm = PASS01.codm (PASS01))

Π medicos.nome, pacientes.nome ((pacientes) [X]
pacientes.codp = PASS02.codp(PASS02))

A álgebra relacional empresta da teoria de conjuntos quatro operadores: União, Intersecção, Diferença e Produto Cartesiano

Sintaxe da **União**: $\langle \text{tabela} \rangle 1 \cup \langle \text{tabela} \rangle 2$

Sintaxe da **Intersecção**: $\langle \text{tabela} \rangle 1 \cap \langle \text{tabela} \rangle 2$

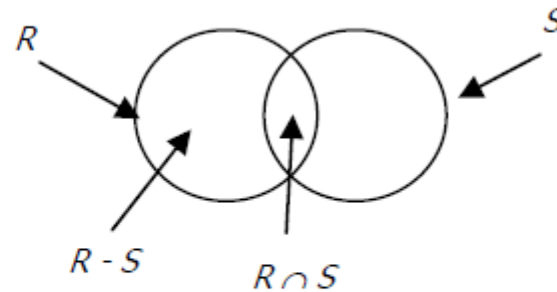
Sintaxe da **Diferença**: $\langle \text{tabela} \rangle 1 - \langle \text{tabela} \rangle 2$

Nos três casos, a operação possui **duas tabelas** como **operando**. E as tabelas devem ser **compatíveis**:

- possuir o mesmo número de colunas;
- o mesmo domínio para cada posição da lista de atributos;
- quando os nomes das colunas forem diferentes, adota-se os nomes das colunas da primeira tabela.

Álgebra Relacional

Operações da teoria dos conjuntos



Nos dois operadores valem as regras apresentadas para a união:

- mesmo número de colunas nos operandos;
- colunas posicionalmente correspondentes com tipos compatíveis.

- ▶ **Notação:** $R \cup S$
 - ▶ R e S devem ter o mesmo esquema
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Contém todas as linhas de R e de S
 - ▶ O esquema é o mesmo das tabelas de entrada
 - ▶ Duplicidade é eliminada

$$C1 \cup C2$$

<u>CliId</u>	Nome
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	Nome
1532	Asdrúbal
1644	Jepeto
1780	Quincas
1982	Zandor

<u>CliId</u>	Nome
1532	Asdrúbal
1644	Jepeto
1755	Doriana
1780	Quincas
1982	Zandor

Selecione todos os empregados que trabalham no departamento número 2 ou que supervisionam empregados que trabalham no departamento número 2.

Ex:

Aluno = { nome, idade, curso}

Professor = { nome, idade, depto.}

Funcionario = { nome, depto, idade}

Dom(nome) = varchar(30)

Dom(idade) = int

Dom(curso) = varchar(5)

Dom(depto) = varchar(5)

**➡ Aluno é compatível com Professor,
mas não é com Funcionario.**

Apresente uma relação com todos os alunos e também com todos os professores:

Aluno = {nome, idade, curso}

{Zeca, 25, comput.

Zico, 21, eletr.

Juca, 19, odonto.

Tuca, 19, comput.}

Professor = {nome, idade, depto.}

{Ari, 35, comput.

Wilma, 32, eletr.

Zeca, 25, comput.}

Aluno \cup Professor = {nome, idade, curso}

{ Zeca, 25, comput.

Zico, 21, eletr.

Juca, 19, odonto.

Tuca, 19, comput.

Ari, 35, comput.

Wilma, 32, eletr. }

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome, idade, cidade de médicos ou pacientes que tenham Silva no nome?

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome, idade, cidade de médicos ou pacientes que tenham Silva no nome?

MEDSILVA $\leftarrow (\Pi \text{ nome, idade, cidade}(\sigma \text{ nome like ' \%Silva\%'(medicos))))$

PACSILVA $\leftarrow (\Pi \text{ nome, idade, cidade}(\sigma \text{ nome like ' \%Silva\%'(pacientes))))$

MEDSILVA \cup **PACSILVA**

- ▶ **Notação:** $R - S$
 - ▶ R e S devem ter o mesmo esquema
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Contém todas as linhas de R e que não são encontradas em S
 - ▶ O esquema é o mesmo das tabelas de entrada

$C1 - C2$

<u>CliId</u>	Nome
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	Nome
1532	Asdrúbal
1644	Jepeto
1780	Quincas
1982	Zandor

<u>CliId</u>	Nome
1755	Doriana

Operação de Diferença

EMPREGADO	NomeP	NomeF	NumBI	...	Salário	SuperBI	NumDep
	João	Santos	798764544	...	2500	487563546	4
	Inês	Pereira	345673451	...	2000	487563546	1
	Rui	Silva	487563546	...	1500	123456789	2
	Ana	Feio	342342324	...	3000	798764544	4

- Obtenha o número do BI dos empregados que trabalham no departamento 4 e que não supervisionam um empregado que trabalha no departamento 4.

$EMP_DEP4 \leftarrow \sigma_{NumDep = 4}(EMPREGADO)$

$RESULT1 \leftarrow \pi_{NumBI}(EMP_DEP4)$

$RESULT2 \leftarrow \pi_{SuperBI}(EMP_DEP4)$

$RESULT \leftarrow RESULT1 - RESULT2$

RESULT	NumBI
	342342324

Apresente uma relação de todos os alunos que **não são** professores

Aluno = {nome, idade, curso}

{Zeca, 25, comput.

Zico, 21, eletr.

Juca, 19, odonto.

Tuca, 19, comput.}

Professor = {nome, idade, depto.}

{Ari, 35, comput.

Wilma, 32, eletr.

Zeca, 25, comput.}

Aluno - Professor = {nome, idade, curso}

{ Zico, 21, eletr.

Juca, 19, odonto.

Tuca, 19, comput.}

Professor - Aluno = {nome, idade, depto.}

{Ari, 35, comput.

Wilma, 32, eletr.}

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome, idade, cidade de médicos que não são pacientes?

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome, idade, cidade de médicos que não são pacientes?

MED $\leftarrow (\Pi \text{ nome, idade, cidade}(\text{medicos}))$

PAC $\leftarrow (\Pi \text{ nome, idade, cidade}(\text{pacientes}))$

MED - PAC

Liste os nomes dos empregados que não têm dependentes?

1 - União $R \cup S$

- é o conjunto das tuplas que estão em R, em S, ou em ambas
 - R e S da mesma aridade
 - nomes dos atributos a especificar

2 - Diferença $R - S$

- tuplas de R que não estão em S
 - R e S da mesma aridade

R		
A	B	C
a	b	c
d	a	f
c	b	d

S		
D	E	F
b	g	a
d	a	f

$R \cup S$		
A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

$R - S$		
A	B	C
a	b	c
c	b	d

- ▶ **Notação:** $R \cap S$
 - ▶ R e S devem ter o mesmo esquema
- ▶ **Entrada:** Tabela (R) e Tabela (S)
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** Contém todas as linhas de R que são encontradas em S também
 - ▶ O esquema é o mesmo das tabelas de entrada
 - ▶ $R - (R - S)$ ou $S - (S - R)$ ou $R \bowtie S$

$$C1 \cap C2$$

<u>CliId</u>	Nome
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	Nome
1532	Asdrúbal
1644	Jepeto
1780	Quincas
1982	Zandor

<u>CliId</u>	Nome
1532	Asdrúbal
1780	Quincas

Apresente uma relação de todos os alunos que **são** professores

Aluno = {nome, idade, curso}

{Zeca, 25, comput.

Zico, 21, eletr.

Juca, 19, odonto.

Tuca, 19, comput.}

Professor = {nome, idade, depto.}

{Ari, 35, comput.

Wilma, 32, eletr.

Zeca, 25, comput.}

Aluno \cap Professor = {nome, idade, curso}

{ Zeca, 25, comput}

Operadores derivados

há operadores de álgebra que são deriváveis de outros. A operação de intersecção é derivável de união e diferença:

$$A \cap B = A - (A - B)$$

EMPREGADO	NomeP	NomeF	NumBI	...	Salário	SuperBI	NumDep
	João	Santos	798764544	...	2500	487563546	4
	Inês	Pereira	345673451	...	2000	487563546	1
	Rui	Silva	487563546	...	1500	123456789	2
	Ana	Feio	342342324	...	3000	798764544	4

- Obtenha o número do BI dos empregados que trabalham no departamento 4 e que supervisionam um empregado que trabalha no departamento 4.

$EMP_DEP4 \leftarrow \sigma_{NumDep = 4}(EMPREGADO)$

$RESULT1 \leftarrow \pi_{NumBI}(EMP_DEP4)$

$RESULT2 \leftarrow \pi_{SuperBI}(EMP_DEP4)$

$RESULT \leftarrow RESULT1 \cap RESULT2$

RESULT	NumBI
	798764544

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome, idade, cidade de médicos são pacientes?

Ambulatórios(nroa, andar, capacidade)

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

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

Consultas(codm, codp, data, hora)

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

Qual é o nome, idade, cidade de médicos são pacientes?

MED $\leftarrow (\Pi \text{ nome, idade, cidade}(\text{medicos}))$

PAC $\leftarrow (\Pi \text{ nome, idade, cidade}(\text{pacientes}))$

MED \cap PAC

Interseção $R \cap S$

- contém as tuplas que pertencem a R e a S simultaneamente
 - R e S da mesma aridade
- $R \cap S = R - (R - S)$

R

A	B	C
a	b	c
d	a	f
c	b	d

S

D	E	F
b	g	a
d	a	f

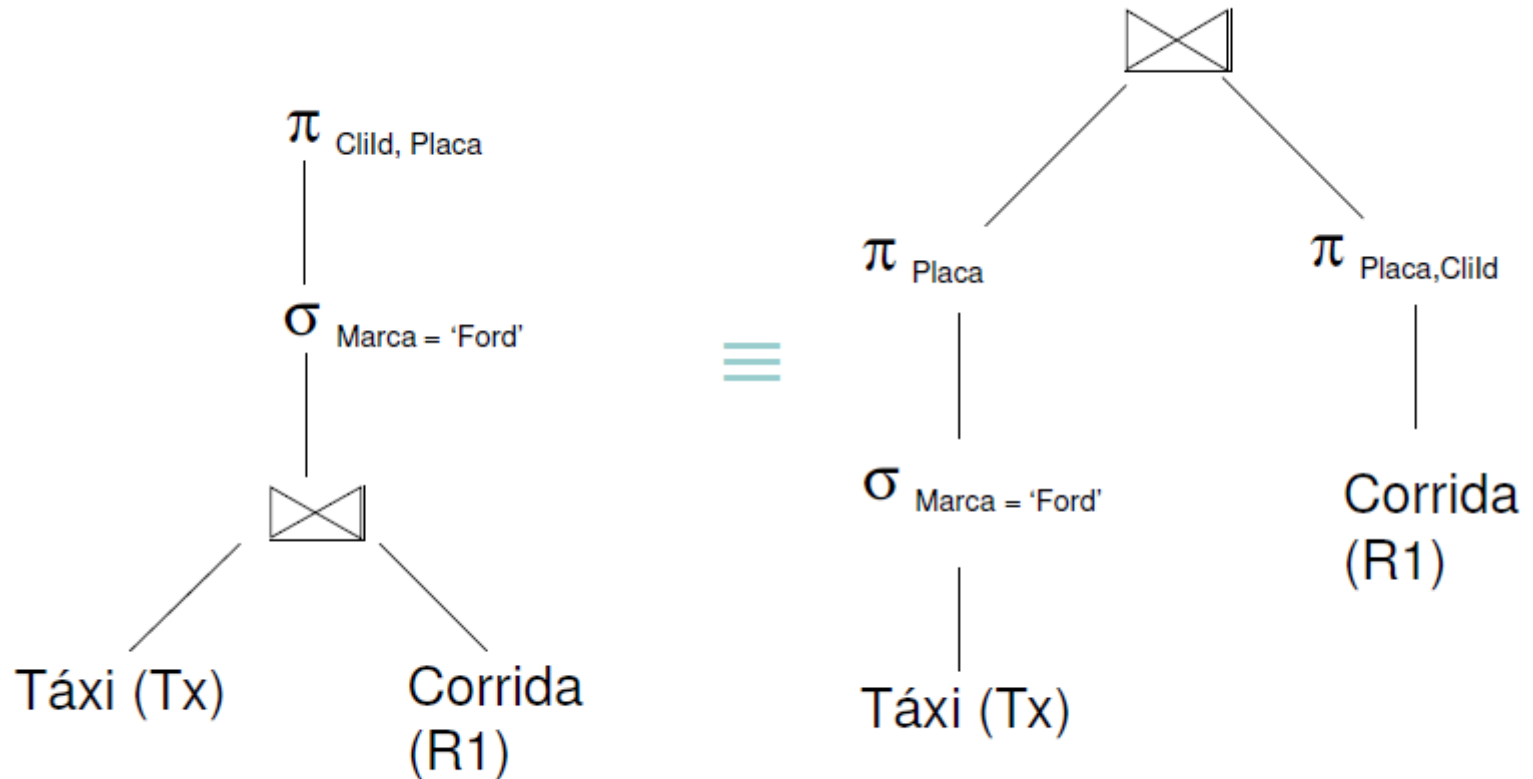
$R \cap S$

A	B	C
d	a	f

- União e interseção são
 - comutativas:
 - $\mathbf{R \cup S = S \cup R}$
 - $\mathbf{R \cap S = S \cap R}$
 - associativas:
 - $\mathbf{R \cup (S \cup T) = (R \cup S) \cup T}$
 - $\mathbf{(R \cap S) \cap T = R \cap (S \cap T)}$
- Diferença não é comutativa
 $\mathbf{R - S \neq S - R}$

Álgebra Relacional

Otimizacao de consultas



- a –** $(R - S) \cup S = R$
- b –** $(R - S) \cup (R \cap S) = R$
- d –** $(R - S) \cup (S - R) = (R \cup S) - (R \cap S)$

Demonstre a resposta com exemplos de conjuntos

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

$$R = \{a, b\}$$

$$S = \{a\}$$

$$R - S = \{b\}$$

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

$$R = \{a, b\}$$

$$S = \{a\}$$

$$R - S = \{b\}$$

$$R \cap S = \{a\}$$

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

$$R = \{a, b\}$$

$$S = \{a\}$$

$$R - S = \{b\}$$

$$S - R = \{\}$$

$$R \cup S = \{a, b\}$$

$$R \cap S = \{a\}$$

- ▶ **Notação:** $R \div S$
- ▶ **Entrada:** Tabela (R) e Tabela (S)
 - ▶ Seja grau a medida de atributos de mesmo nome
 - ▶ R tem grau (“m”+”n”)
 - ▶ S tem grau “n”
- ▶ **Propósito:** gera linhas de acordo com um critério
- ▶ **Saída:** atributos de S cujos valores associam-se com todos os valores de R
 - ▶ Grau “m”

- Encontre clientes que tenham andado em todos os táxis da Marca Ford

CIId	Placa
1532	DAE6534
1532	DKL4586
1644	DKL7878
1644	JDM8776
1780	JJM3692
1982	DAE6534
1982	DKL4598
1982	DKL7878

$\rho(\text{SR2}, \pi_{\text{CIId}, \text{Placa}}(\text{R2}))$



$\rho(\text{SFR}, \pi_{\text{Placa}}(\text{FR}))$



Placa
DAE6534
DKL7878

- Procurar em uma dada relação as “subtuplas” que são completadas por TODAS as tuplas de outra relação
- $Q(A_1, A_2, \dots, A_p) = R(A_1, A_2, \dots, A_p, A_{p+1}, \dots, A_n) / S(A_{p+1}, \dots, A_n)$
- Q contém tuplas tais que, concatenadas à todas as tuplas de S, formam tuplas de R.

R	A	B
	a1	b1
	a2	b1
	a3	b1
	a4	b1
	a1	b2
	a3	b2
	a2	b3
	a3	b3
	a4	b3
	a1	b4
	a2	b4
	a3	b4

S	A
	a1
	a2
	a3

$R \div S$	B
	b1
	b4

- Obtenha o nome dos empregados que trabalham em TODOS os projectos nos quais o Rui Silva também trabalha.

$$\text{EMP_SILVA} \leftarrow \sigma_{\text{NomeP} = \text{'Rui'} \text{ AND } \text{NomeF} = \text{'Silva'}}(\text{EMPREGADO})$$
$$\text{SILVA_PROJ} \leftarrow \pi_{\text{NumProj}}(\text{TRABALHA_EM} \bowtie_{\text{EmpBI} = \text{NumBI}} \text{EMP_SILVA})$$
$$\text{BI_PROJ} \leftarrow \pi_{\text{EmpBI}, \text{NumProj}}(\text{TRABALHA_EM})$$
$$\text{RESULT_BI} \leftarrow \rho_{(\text{NumBI})}(\text{BI_PROJ} \div \text{SILVA_PROJ})$$
$$\text{RESULT} \leftarrow \pi_{\text{NomeP}, \text{NomeF}}(\text{RESULT_BI} * \text{EMPREGADO})$$

É uma operação adicional que produz como resultado a projeção de todos os elementos da primeira tabela que se relacionam com todos os elementos da segunda tabela.

Ex.: queremos saber os nomes dos departamentos que possuem todos os cargos:

$$\pi_{NmDeppto, CdCargo} (\text{depto} \mid x \mid \text{funcionário}) \div \pi_{CdCargo} (\text{cargo})$$

- **Operação de Divisão:** a divisão de duas relações $R \div S$, onde os atributos de S estão contidos nos atributos de R , $A(R) \supseteq A(S)$, resulta na relação T , onde $A(T) = A(S)$, onde para cada tupla t que aparece no resultado, os valores de t devem aparecer em R , combinando com cada tupla de S . Esta operação é utilizada nas consultas em que se emprega a frase "para todos";

S
A1
a1
a2
a3

$R \div S$
A2
b1
b4

R	A2
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

Álgebra Relacional divisão

- **Ex:** encontrar todos os clientes que tenham conta em todas as agências localizadas em São Leopoldo.

conta		
nome_agencia	numero_conta	saldo
SAL-1	0001	1200
SAL-1	0002	3000
NÔH-1	0003	4500
PÔA-1	0004	4000
PÔA-1	0005	1500
NÔH-1	0006	200
SAL-2	0007	3750
SAL-2	0008	1800

agencia		
nome_agencia	cidade_agencia	saldo
NÔH-1	Novo Hamburgo	260050
SAL-1	São Leopoldo	455580
PÔA-1	Porto Alegre	1250369
SAL-2	São Leopoldo	125588

depositante	
nome_cliente	numero_conta
João	0001
Pedro	0002
Francisco	0003
Maria	0004
Paulo	0007
José	0006
Ana	0005
João	0008

$\pi_{\text{nome_agencia}}(\sigma_{\text{cidade_agencia} = \text{"São Leopoldo"}}(\text{agencia}))$

nome_agência	cidade_agência	saldo
SAL-1	São Leopoldo	455580
SAL-2	São Leopoldo	125588

⋈

nome_agência
SAL-1
SAL-2

$\pi_{\text{nome_cliente, nome_agencia}}(\text{depositante} \bowtie \text{conta})$

nome_cliente	nome_agencia
João	SAL-1
Pedro	SAL-1
Francisco	NÔH-1
Maria	PÔA-1
Paulo	SAL-2
José	NÔH-1
Ana	PÔA-1
João	SAL-2

$\pi_{\text{nome_cliente, nome_agencia}}(\text{depositante} \bowtie \text{conta}) \div$
 $\pi_{\text{nome_agencia}}(\sigma_{\text{cidade_agencia} = \text{"São Leopoldo"}}(\text{agencia}))$

nome_cliente
João

R_1	R_2a	R_2b	R_2c	$R_1 \dot{-} R_2a$	$R_1 \dot{-} R_2b$																																						
<table> <tr><th>x</th><th>y</th><th>z</th></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>2</td><td>2</td></tr> <tr><td>3</td><td>1</td><td>3</td></tr> </table>	x	y	z	1	1	1	1	2	1	2	1	1	2	2	2	3	1	3	<table> <tr><th>z</th></tr> <tr><td>1</td></tr> </table>	z	1	<table> <tr><th>y</th><th>z</th></tr> <tr><td>1</td><td>1</td></tr> </table>	y	z	1	1	<table> <tr><th>y</th></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> </table>	y	1	2	<table> <tr><th>x</th><th>y</th></tr> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>2</td></tr> <tr><td>2</td><td>1</td></tr> </table>	x	y	1	1	1	2	2	1	<table> <tr><th>x</th></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> </table>	x	1	2
x	y	z																																									
1	1	1																																									
1	2	1																																									
2	1	1																																									
2	2	2																																									
3	1	3																																									
z																																											
1																																											
y	z																																										
1	1																																										
y																																											
1																																											
2																																											
x	y																																										
1	1																																										
1	2																																										
2	1																																										
x																																											
1																																											
2																																											
					$R_1 \dot{-} R_2c$																																						
					<table> <tr><th>x</th><th>z</th></tr> <tr><td>1</td><td>1</td></tr> </table>	x	z	1	1																																		
x	z																																										
1	1																																										

Divisão R / S

- aridade de R é r e de S é s, $r > s$, $S \neq \emptyset$
- contém as $(r-s)$ -tuplas (a_1, \dots, a_{r-s}) tais que, para **todas** as s-tuplas (a_{r-s+1}, \dots, a_r) em S, a tupla (a_1, \dots, a_r) está em R

R				S		R/S	
A	B	C	D	E	F	A	B
a	b	c	d	c	d	a	b
a	b	e	f	e	f	e	d
b	c	e	f				
e	d	c	d				
e	d	e	f				
a	b	d	e				

$$\begin{array}{r|l} 7 & 3 \\ \hline 1 & 2 \end{array}$$

R/S		S	
A	B	E	F
a	b	c	d
e	d	e	f

$$2 \times 3 + 1 = 7$$

- Forma de proceder à divisão
 - reordenar as colunas de forma a que as últimas correspondam ao quociente
 - ordenar a tabela pelas primeiras colunas
 - cada subtupla das primeiras colunas pertence ao resultado se o conjunto de subtuplas das últimas colunas que lhe corresponde contiver o quociente

R				S		R/S	
A	B	C	D	E	F	A	B
a	b	c	d	c	d	a	b
		e	f	e	f	e	d
		d	e				
b	c	e	f				
e	d	c	d				
		e	f				

Álgebra Relacional

Divisão

EQUIPE

ID_EMP	COD_PROJ
17206-2	001
12584-7	002
16764-6	001
17206-2	002
15698-3	003
17206-2	003

PROJETOS

COD_PROJ	DESCRICAO
001	Sistema IRPF
002	Sistema RH
003	Sistema Banco

FUNCIONARIO

ID_EMP	NOME	CARGO
17206-2	Jorge	Analista
12584-7	Paula	Programadora
16764-6	Frederico	DBA
15698-3	Heloisa	Web Master

Imagine a situação de querermos saber quais os funcionários que trabalham em todos os projetos:

RESULT <----- $(\pi_{\text{COD_PROJ}}(\text{PROJETOS})) \div (\pi_{\text{ID_EMP, COD_PROJ}}(\text{EQUIPE}))$

RESULT

ID_EMP
17206-2

- x e y podem ser quaisquer listas de campos
 - $x \cup y$ é a lista de campos na relação A
 - y é a lista de campos na relação B
- A/B
 - $A/B = \{ \langle x \rangle \mid \exists \langle x, y \rangle \in A \ \forall \langle y \rangle \in B \}$
 - A/B contém todas tuplas x (clientes) tal que para cada tupla y (táxi de marca MA1) em B , há uma tupla xy em A .
 - Ou: Se o conjunto de valores y (marca do táxi) associados com valores x (clientes) em A contiverem todos os valores y em B , o valor x está em A/B .

X	Y
x1	y1
X1	y2
X1	y3
X1	y4
x2	y1
x2	y2
x3	y2
X4	y2
x4	y4

A

Y
y2

B1

Y
y2
y4

B2

Y
y1
y2
y4

B3

X
x1
x2
x3
x4

A/B1

X
x1
x4

A/B2

X
x1

A/B3

TY

Placa	Marca	Modelo	Ano
AAA0101	Ma1	Mo1	2000
FFF0606	Ma1	Mo2	2004

Consulta:
*Encontrar clientes que tenham
andado em táxis da marca Ma1*

Corrida (C)

CliId	Placa	Data
0101	AAA0101	10/01/2001
0101	FFF0606	
0202	FFF0606	
0202	BBB0202	
0303	DDD0404	
0404	AAA0101	
0404	FFF0606	19/02/2002
0404	DDD0404	20/02/2002

Corrida2

CliId	Placa
0101	AAA0101
0101	FFF0606
0202	FFF0606
0202	BBB0202
0303	DDD0404
0404	AAA0101
0404	FFF0606
0404	DDD0404

$\rho(\text{Corrida2}, \pi_{\text{CliId}, \text{Placa}}(\text{Corrida}))$

$\rho(\text{Taxi2}, \pi_{\text{Placa}}(\text{TY}))$

Taxi2

Placa
AAA0101
FFF0606

Corrida2

CliId	Placa
0101	AAA0101
0101	FFF0606
0202	FFF0606
0202	BBB0202
0303	DDD0404
0404	AAA0101
0404	FFF0606
0404	DDD0404

Taxi2

Placa
AAA0101
FFF0606

Corrida2/Taxi2

CliId
0101
0404

- $F(\text{numfd}, \text{nomef})$
- $D(\text{numfd}, \text{nomed}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais os nomes e parentescos de todos os dependentes?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais os nomes e parentescos de todos os dependentes?

■ $\pi_{\text{nomed}, \text{par}} (D)$

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários possuem dependentes filhas?

■ $\pi_{\text{numfd}} (\sigma_{\text{par}='filha'} (D))$

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários não possuem dependentes?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \text{nomed}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários não possuem dependentes?

$$\blacksquare \pi_{\text{numfd}}(F) - \pi_{\text{numfd}}(D)$$

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Dê os nomes dos funcionários que possuem algum dependente.
- Dê o nome de cada funcionário que possui uma dependente chamada Alice.
- Quais funcionários não têm Alice como dependente?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Dê os nomes dos funcionários que possuem algum dependente.
 - $\pi_{\text{nomef}} (F \bowtie D)$

- $F(\text{numfd}, \text{nomef})$
- $D(\text{numfd}, \text{nomed}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Dê o nome de cada funcionário que possui uma dependente chamada Alice.

■ $\pi_{\text{nomef}} (F \mid X | (\sigma_{\text{nomed}='Alice'} (D)))$

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
numfd	nomef
01	F1
02	F2
03	F3
04	F4

D		
numfd	nomed	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários não têm Alice como dependente?
 - $\pi_{\text{numfd}}(D) - \pi_{\text{numfd}}(\sigma_{\text{nomed}='Alice'}(D))$

branch (branch-name, branch-city, assets)

*customer (customer-name, customer-street,
customer-only)*

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)

- Determinar todos os empréstimos superiores a \$1200
- Encontrar os números dos empréstimos de montante superior a \$1200
- Listar os nomes de todos os clientes que têm um empréstimo, uma conta, ou ambas as coisas
- Encontrar os clientes que têm um empréstimo e uma conta no banco.
- Determinar todos os clientes que têm um empréstimo na agência de Perryridge.
- Listar os nomes dos clientes que possuem um empréstimo na agência de Perryridge mas que não tem nenhuma conta no banco.
- Determinar todos os clientes que têm um empréstimo na agência de Perryridge.
- Determinar o saldo mais elevado entre todas as contas

médicos(nEmpr, nomeM, especialidade)

pacientes(nBI, nomeP, telefone, morada, idade)

fármacos(codF, nomeF)

consultas(nConsulta, data, nBI, nEmpr)

receitas(codF, nConsulta, quantidade)

- ❑ Quais os pacientes com mais de 50 anos de idade?
- ❑ Quais os nomes dos pacientes com mais de 50 anos de idade?
- ❑ Quais os fármacos que já foram receitados em consultas da clínica?
- ❑ Quais os fármacos que nunca foram receitados?
- ❑ Qual a idade do paciente mais velho?
- ❑ E quais os (nomes dos) pacientes com essa idade?

Álgebra Relacional

Expressão Quociente

- $T = \Pi_{1, \dots, r-s} (R)$ = universo das tuplas possíveis no resultado
- $W = (T \times S) - R$ = todas as linhas T combinadas com S mas que não estão em R, i.e., em que a condição falha
- $V = \Pi_{1, \dots, r-s} (W)$ = tuplas que não interessam
- $R / S = T - V$ = tuplas que interessam
 - reunindo numa só expressão algébrica

$$R / S = \Pi_{1, \dots, r-s} (R) - \Pi_{1, \dots, r-s} [(\Pi_{1, \dots, r-s} (R) \times S) - R]$$

T	
A	B
a	b
b	c
e	d

S	
E	F
c	d
e	f

T × S			
A	B	E	F
a	b	c	d
a	b	e	f
b	c	c	d
b	c	e	f
e	d	c	d
e	d	e	f

W			
A	B	E	F
b	c	c	d

V	
A	B
b	c

R/S	
A	B
a	b
e	d

- ▶ Para aquelas consultas que não podem ser resolvidas simplesmente através da álgebra relacional, introduz-se um conjunto de funções agregadas
- ▶ Funções comumente aplicadas a conjuntos de dados são: Média, Máximo, Mínimo, Soma, Contador

Álgebra Relacional

Funções Agregadas

• **Funções agregadas:** são aquelas que, quando aplicadas, tomam uma coleção de valores e retornam um valor simples como resultado.

- **função sum:** descobrir a soma total dos salários de todos os empregados de tempo integral.

`sum_salario (trabalhador_integral)`

sum salario
14505

trabalhador_integral		
nome_empregado	nome_agencia	salario
José	NQH-1	5000
Ana	POA-1	4800
Flávia	SAL-1	3200
Maria	POA-1	6500

- **função count:** descobrir o número de agências existentes na tabela de tempo integral.

`count_nome_agencia (trabalhador_integral)`

count nome_agencia
4

`count-distinct_nome_agencia (trabalhador_integral)`

count-distinct nome_agencia
3

- **função avg:** descobrir a média dos salários.

`avg_salario (trabalhador_integral)`

avg salario
3626.25

- **função min:** descobrir o menor salário.

`min_salario (trabalhador_integral)`

min salario
3200

- **função max:** descobrir o maior salário.

`max_salario (trabalhador_integral)`

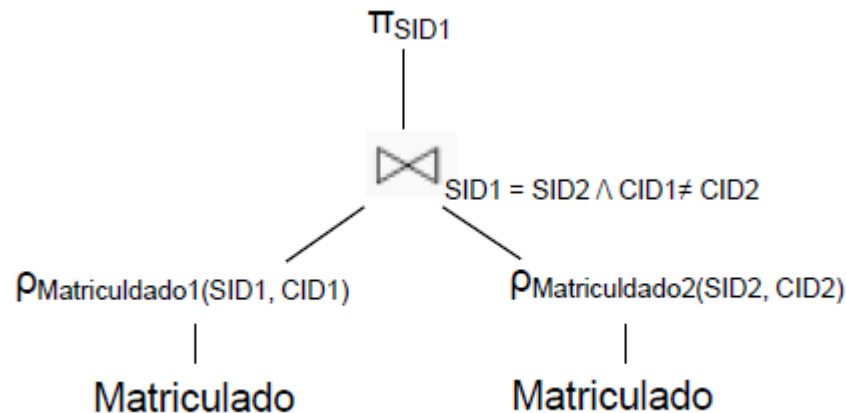
max salario
6500

<i>Símbolo</i>	<i>Operação</i>	<i>Sintaxe</i>	<i>Tipo</i>
σ	Seleção / Restrição	$\sigma_{\text{condição}} (\text{Relação})$	Primitiva
π	Projeção	$\pi_{\text{expressões}} (\text{Relação})$	Primitiva
\cup	União	$\text{Relação1} \cup \text{Relação2}$	Primitiva
\cap	Intersecção	$\text{Relação1} \cap \text{Relação2}$	Adicional
-	Diferença de conjuntos	$\text{Relação1} - \text{Relação2}$	Primitiva
x	Produto cartesiano	$\text{Relação1} \times \text{Relação2}$	Primitiva
 x 	Junção	$\text{Relação1} \bowtie \text{Relação2}$	Adicional
\div	Divisão	$\text{Relação1} \div \text{Relação2}$	Adicional
ρ	Renomeação	$\rho_{\text{nome}} (\text{Relação})$	Primitiva
\leftarrow	Atribuição	$\text{variável} \leftarrow \text{Relação}$	Adicional

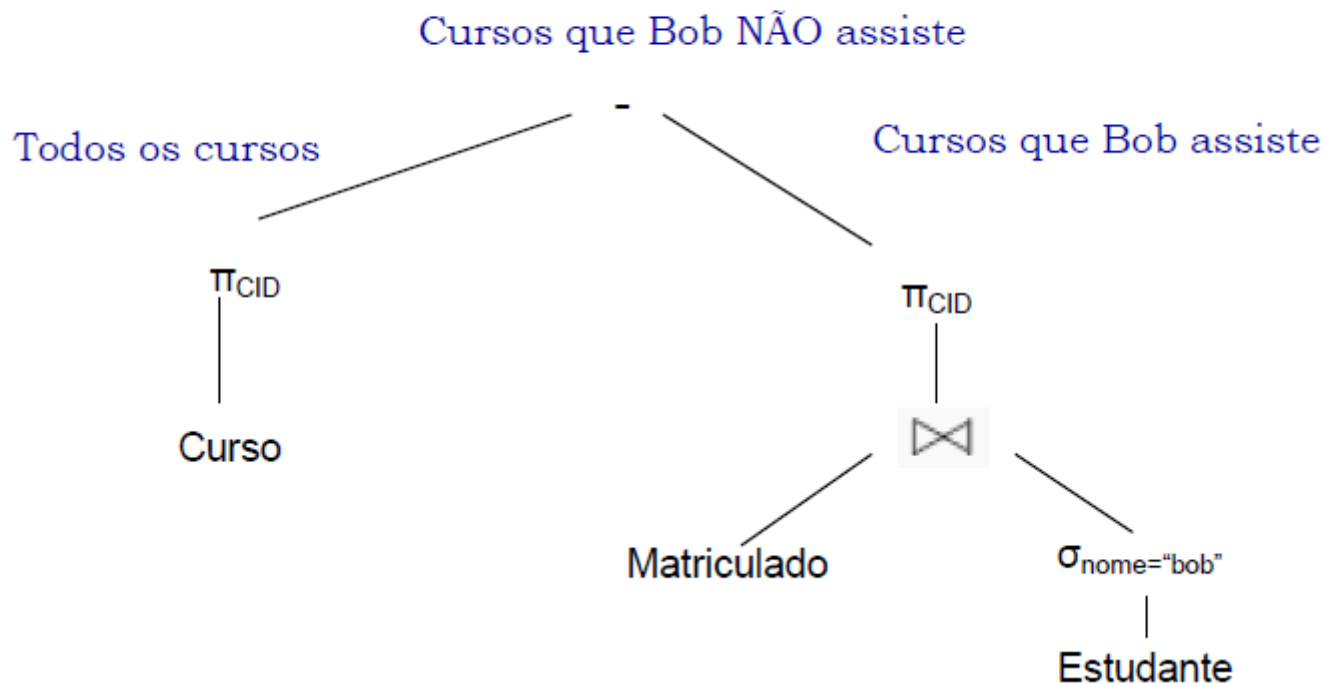
SÍMBOLO	OPERAÇÃO	SINTAXE	TIPO
\leftarrow	Atribuição	Variável \leftarrow Relação	Primitiva
σ	Seleção (Select)	σ <condicao de elecao>(Relação)	Primitiva
π	Projeção (Project)	π <lista de atributos>(Relação)	Primitiva
\cup	União (Union)	(Relação 1) \cup (Relação 2)	Primitiva
\cap	Interseção (Intersection)	(Relação 1) \cap (Relação 2)	Adicional
$-$	Diferença (Difference)	Sintaxe: (Relação 1) $-$ (Relação 2)	Primitiva
\times	Produto Cartesiano (Product)	(Relação 1) \times (Relação 2)	Primitiva
\bowtie	Junção (Join)	(Relação 1) \bowtie <condição de junção> (Relação 2)	Adicional
\div	Divisão (Divide)	(Relação 1) \div (Relação 2)	Adicional

- ▶ Estud (SID, nome, idade, CR)
- ▶ Curso (CID, titulo)
- ▶ Matr (SID, CID)

- ▶ SID dos estudantes que assistem a pelo menos 2 cursos
- ▶ $r1 \leftarrow \text{SELECIONE Matric COM SID=SID1 E CID} = \text{CID1}$
- ▶ $r2 \leftarrow \text{SELECIONE Matric COM SID=SID2 E CID} = \text{CID2}$
- ▶ $r3 \leftarrow \text{JUNTE } r1 \text{ A } r2 \text{ COM } r1.\text{SID}=r2.\text{SID E } r1.\text{CID} \neq r2.\text{CID}$
- ▶ $r4 \leftarrow \text{PROJETE } r3 \text{ SOBRE SID}$

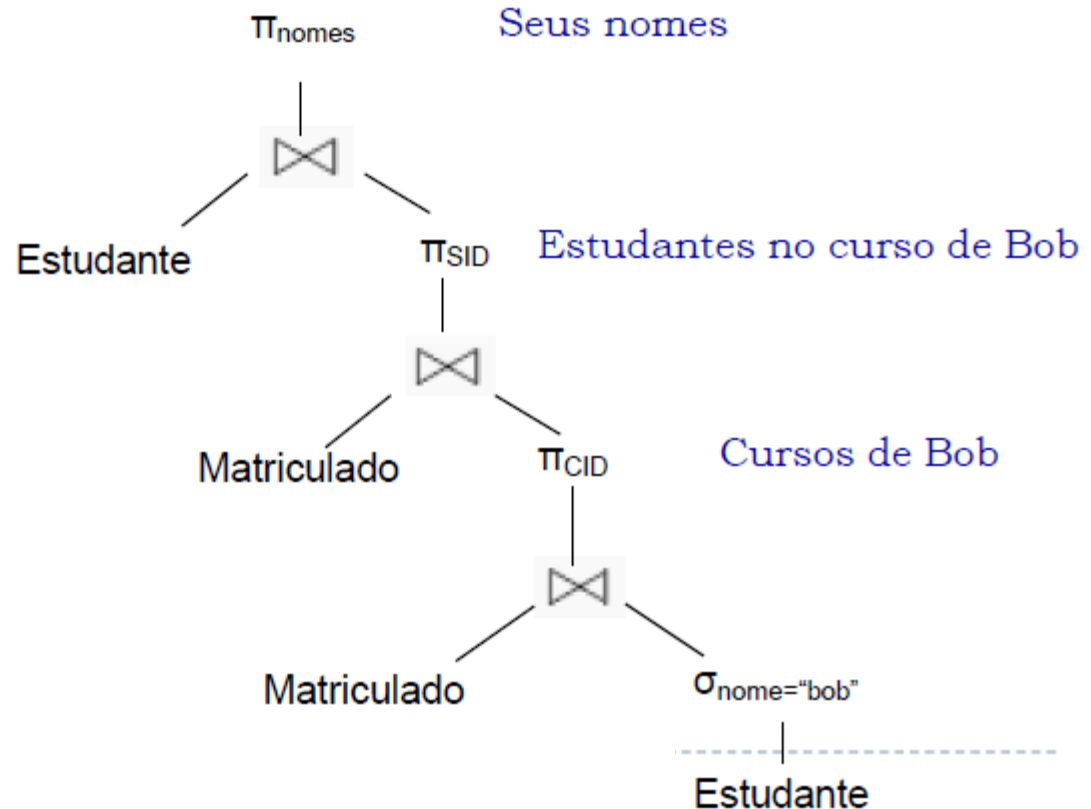


- Listar todos os cursos que Bob NÃO assiste



- ▶ Listar todos os cursos que Bob NÃO assiste
- ▶ $r1 \leftarrow \text{SELECIONE Estud COM nome='Bob'}$
- ▶ $r2 \leftarrow \text{JUNTE } r1 \text{ A Matric COM } r1.CID = \text{Matric.CID}$
- ▶ $r3 \leftarrow \text{PROJETE } r2 \text{ SOBRE CID}$
- ▶ $r4 \leftarrow \text{PROJETE Curso SOBRE CID}$
- ▶ $r5 \leftarrow r4 - r3$

- Listar nomes dos estudantes na classe de Bob



- Listar nomes dos estudantes na classe de Bob
- $r1 \leftarrow \text{SELECIONE Estud COM nome='Bob'}$
- $r2 \leftarrow \text{JUNTE } r1 \text{ A Matric COM } r1.CID = \text{Matric.CID}$
- $r3 \leftarrow \text{PROJETE } r2 \text{ SOBRE CID}$
- $r4 \leftarrow \text{JUNTE } r3 \text{ A Matric COM } r3.CID = \text{Matric.CID}$
- $r5 \leftarrow \text{PROJETE } r4 \text{ SOBRE CID}$
- $r6 \leftarrow \text{JUNTE } r5 \text{ A Estud COM } r5.SID = \text{Estud.SID}$
- $r7 \leftarrow \text{PROJETE } r6 \text{ SOBRE nomes}$

8 - θ -Junção $R \bowtie_{i\theta j} S$

- se a aridade de R for r e a de S , s a aridade da θ -junção é $r+s$
- contém as tuplas do produto cartesiano de R por S tais que o componente i está na relação θ com o componente $r+j$ (i.e., o correspondente ao j em S).

○ Expressão da θ -junção

- $R \bowtie_{i\theta j} S = \sigma_{\theta_{i\theta j}}(R \times S)$
 - se θ for $=$, a operação designa-se equijunção
 - $(7, 8, 9)$ é uma tupla pendente de R pois não aparece na θ -junção

R		
A	B	C
1	2	3
4	5	6
7	8	9

S	
D	E
3	1
6	2

$$R \bowtie_{2 \leq 1} S = R \bowtie_{B \leq D} S$$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

9 - Junção natural $R \bowtie S$

- só é aplicável se os componentes das tuplas em R e S forem designados por atributos.
- a operação implícita na junção natural é a igualdade dos atributos com o mesmo nome.
- cada par de atributos iguais dá origem a um único atributo, com o mesmo nome, no resultado

□ expressão: $R \bowtie S = \Pi_{i_1, \dots, i_m} (\sigma_{R.A_1=S.A_1 \wedge \dots \wedge R.A_k=S.A_k} (R \times S))$

- k é o número de atributos comuns a R (aridade r) e S (aridade s) e $m = r + s - k$

tupla pendente \Rightarrow

R			S			$R \bowtie S$			
A	B	C	B	C	D	A	B	C	D
a	b	c	b	c	d	a	b	c	d
d	b	c	b	c	e	a	b	c	e
b	b	f	a	d	b	d	b	c	d
c	a	d				d	b	c	e
						c	a	d	b

- ❑ tuplas pendentes, isto é desemparelhadas, quer em R quer em S, desaparecem na θ -junção e na junção natural
- ❑ a **junção externa** (θ - ou natural) inclui as tuplas pendentes de R ou S completados a nulos
 - $(7, 8, 9)$ é uma tupla pendente de R pois não aparece na θ -junção
 - (b, b, f) idem, na junção natural

$R \bowtie_{B < D}^+ S$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2
7	8	9	⊥	⊥

$R \bowtie S$

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b
b	b	f	⊥

10 - Semi-junção $R \bowtie S$

- projecção nos atributos de R da junção natural de R e S
- $R \bowtie S = \Pi_R(R \Join S)$
 - R em Π_R representa os atributos de R (o seu esquema); em $R \Join S$ R representa a relação (a instância)
- outra expressão: $R \bowtie S = R \Join \Pi_{R \cap S}(S)$
- dá as tuplas de R que têm par em S

R			S			$R \bowtie S$		
A	B	C	B	C	D	A	B	C
a	b	c	b	c	d	a	b	c
d	b	c	b	c	e	d	b	c
b	b	f	a	d	b	c	a	d
c	a	d						

- ❑ na junção natural e na semi-junção os atributos são importantes; para os tornar explícitos escreve-se $R(A_1, \dots, A_n)$
- ❑ é possível renomear colunas e fazer junções naturais como:

S = S(B,C,D)

B	C	D
b	c	d
b	c	e
a	d	b

S(E,F,G) \bowtie S(G,H,I)

E	F	G	H	I
a	d	b	c	d
a	d	b	c	e

- ❑ uma junção natural entre duas relações sem atributos comuns redunda num produto cartesiano porque, após este, não há nenhuma selecção a fazer (equivalente a fazer uma selecção com a condição *True*)
- ❑ **$R(A,B,C) \bowtie S(G,H,I) = R \times S$**

○ Reunião

- associativa: $R \cup (S \cup T) = (R \cup S) \cup T$
- comutativa: $R \cup S = S \cup R$

○ Produto cartesiano

- associativo: $R \times (S \times T) = (R \times S) \times T$
- não comutativo: $R \times S \neq S \times R$

○ Junção natural

- associativa e comutativa (independência da ordem das colunas devida aos atributos): $R \bowtie S = S \bowtie R$
- Por isso \bowtie generaliza facilmente: $R = R_1 \bowtie \dots \bowtie R_n$
 - R contém os tuplos μ tais que, para $1 \leq i \leq n$, μ restringido aos atributos de R_i é um tuplo de R_i

○ θ - junção

- não é comutativa mas é associativa (no caso de os índices serem válidos)

$$R \underset{i\theta_1j}{\bowtie} (S \underset{k\theta_2l}{\bowtie} T) = (R \underset{i\theta_1j}{\bowtie} S) \underset{(r+k)\theta_2l}{\bowtie} T$$

- Álgebra Relacional pode ser usada como linguagem de interrogação à BD
- *P1 - Relativamente à BD “Cursos”, quais os nomes dos professores do 12º grupo?*

$\Pi_{\text{nome}} [\sigma_{\text{grupo} = '12'} (\text{Professor})]$

- *P2 - Quais os nomes e datas de nascimento dos alunos do curso ‘CG1’ nascidos antes de 1983?*

$\Pi_{\text{nome}, \text{data_nasc}} [\sigma_{\text{codcurso} = 'CG1' \wedge \text{data_nasc} < 1983-01-01} (\text{Aluno})]$

○ *P3 - Nomes dos alunos inscritos à disciplina 327?*

- ❑ nenhuma relação contém nomes de alunos e códigos de disciplina
- ❑ mas a junção $\text{Aluno} \bowtie \text{Inscrito} = R$ contém:
- ❑ $R(\underline{\text{bia}}, \text{nome}, \text{morada}, \text{telefone}, \text{data_nasc}, \text{codcurso}, \text{ano}, \text{letra}, \text{coddiss}, \text{resultado})$

(i) - $\Pi_{\text{nome}} [\sigma_{\text{coddiss} = 327} (\text{Aluno} \bowtie \text{Inscrito})]$

(ii) - $\Pi_{\text{nome}} [\text{Aluno} \bowtie \sigma_{\text{coddiss} = 327} (\text{Inscrito})]$

- ❑ esta maneira de ligar informações no modelo de dados dá muita liberdade para exprimir perguntas arbitrárias mas exige uma fase de optimização para executar (ii) mesmo que a pergunta seja (i)

Núcleo da álgebra relacional : Π, σ, \bowtie

○ renomeação de atributos

$$\square R(A, B, C)$$

$$\square R'(X, Y, Z) = \Pi_{X=A, Y=B, Z=C} [R(A, B, C)]$$

- onde não houver ambiguidades, a simples menção dos atributos, em conjunto com o nome da relação, faz a renomeação

OU

$$- R'(X, Y, Z) = R(A, B, C)$$

OU

$$- R' = \Pi_{X=A, Y=B, Z=C} (R)$$

- expressões aritméticas (+, -, *, /)

$$\square \text{ este mecanismo serve para dar nomes a expressões}$$

$$\square S = \Pi_{W=A * B - C, U=C/B, A} (R)$$

OU

$$\square S(W, U, A) = \Pi_{A * B - C, C/B, A} (R)$$

○ P4 - *Obtenha uma relação de inscrições com as classificações inflacionadas de 20%.*

- $\Pi_{\text{coddis, bia, resultado, novo} = \text{resultado} * 1.2} (\text{Inscrito})$
- $\Pi_{\text{coddis, bia, resultado, novo} = \text{resultado} + (20 - \text{resultado}) / 10} (\text{Inscrito})$

○ nos parâmetros da projecção:

- no membro direito só podem ser usados nomes de atributos do argumento de Π ; no esquerdo só pode estar um atributo (novo...)

○ Operadores de agregação

- CNT (contagem), SUM (adição), AVG (média), MAX (máximo), MIN (mínimo)

○ $S = \Pi_{V = \text{CNT}(B)}(R)$

- $S(V)$ tem um único valor, o número de tuplas de R com valor não nulo no atributo B ($\text{CNT} (*)$ conta todas as linhas) \rightarrow toda a relação agregada

○ $T = \Pi_{A, M = \text{MAX}(B)}(R)$

- $T(A, M)$ tem tantos pares quantos os valores diferentes de A , sendo indicado para cada A o respectivo valor máximo de B (não nulo ...);
- é feita uma partição segundo os atributos de projecção sem operadores de agregação e cada classe é agregada numa só tupla;
- é possível misturar agregações e aritmética

○ P5 - *Qual o número de inscrições, nota média das inscrições, soma de todas as notas e número de resultados não nulos ?*

- $R(NI, M, T, NR) = \Pi_{CNT(*), AVG(resultado), SUM(resultado), CNT(resultado)}$
(Inscrito)
- pode ser $M \neq T/NI$ se houver inscrições ainda sem resultado (valor nulo); tem que ser $M = T / NR$

Inscrito

codd	dis	bia	resultado
PA	97	14	
PA	38	12	
ITI	97	17	
ITI	25	14	
H	97	10	
H	25		

R

NI	M	T	NR
6	13.4	67	5

Cliente Particular (CP)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Cliente Empresa (CE)

<u>CliId</u>	Nome	CGC
1532	Asdrúbal	754.856.965/0001-54
1644	Jepeto	478.652.635/0001-75
1780	Quincas	554.663.996/0001-87
1982	Zandor	736.952.369/0001-23

Táxi (TX)

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolksvagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolksvagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003



- Qual a marca de carro mais requisitada pelos clientes?
- Em que mês do ano mais corridas são feitas?
- Qual o nome dos clientes que trabalham em uma determinada empresa que mais se utiliza do serviço de táxi?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	<u>nomef</u>
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais os nomes e parentescos de todos os dependentes?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	<u>nomef</u>
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários possuem dependentes filhas?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	nomef
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários não possuem dependentes?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	nomef
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Dê os nomes dos funcionários que possuem algum dependente.

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	nomef
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Dê o nome de cada funcionário que possui uma dependente chamada Alice.

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	nomef
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários não têm Alice como dependente (isto é, nenhuma dependente chamada Alice)?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	<u>nomef</u>
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários possuem mais de um dependente?

- $F(\underline{\text{numfd}}, \text{nomef})$
- $D(\underline{\text{numfd}}, \underline{\text{nomed}}, \text{par})$

F	
<u>numfd</u>	nomef
01	F1
02	F2
03	F3
04	F4

D		
<u>numf</u>	<u>nomed</u>	par
01	Alice	filha
02	Alice	esposa
02	Clara	filha
03	José	filho

- Quais funcionários possuem mais de um dependente?
 - $\rho_{D_1}(\text{numf}, \text{nomed1}, \text{par1}) (D)$
 - $\rho_{D_2}(\text{numf}, \text{nomed2}, \text{par2}) (D)$
 - $\pi_{\text{numf}}((\sigma_{\text{nomed1} \neq \text{nomed2}}(D_1 \times D_2)))$

- **Formato:** $\text{RelResultado} = \sigma_{\text{predicado}} (\text{RelEntrada})$
- Ex.: Selecione as tuplas da relação EMPRÉSTIMOS para quais o nome da agência é “Perryridge”

$$R = \sigma_{e\text{-agência}='Perryridge'} (\text{EMPRÉSTIMOS})$$

```
SELECT *  
FROM EMPRESTIMOS  
WHERE E_AGENCIA = 'PERRYRIDGE'
```

- Formato: $\text{RelResult} = \pi_{\text{colunas a copiar}}(\text{RelEntrada})$
- Ex.: Obter uma tabela que relacione os clientes do banco com as agências onde fizeram empréstimos:

$$\text{Res} = \pi_{\text{e-agencia, e-nome}}(\text{EMPRÉSTIMOS})$$

```
SELECT E_AGENCIA, E_NOME  
FROM EMPRESTIMOS
```


- É possível compor operações mais complexas da álgebra relacional através do aninhamento de operações mais simples
- Exemplo: listar os nomes dos clientes que fizeram empréstimos superiores a 1200:

$$RelResult = \pi_{e-nome}(\sigma_{e-valor > 1200}(EMP\acute{R}ESTIMOS))$$

```
SELECT DISTINCT E_NOME
FROM (SELECT E_NOME
      FROM EMPRESTIMOS
      WHERE E_VALOR > 1200)
```

- A operação de seleção que gera uma relação como resultado pode ser usada como relação de entrada para a operação de projeção

Em SQL proporciona um mecanismo para renomear tanto atributos quanto relações, usando a cláusula **as**, como segue:

```
select Atributo as Novo_nome  
from relação;
```

- Exemplo: listar o nome dos clientes que moram em Rye e fizeram empréstimo de menos de 1000.
 - a) $\text{RelResult1} = \text{CLIENTES} \times \text{EMPRÉSTIMOS}$
 - b) $\text{RelResult2} = \sigma_{c\text{-name} = e\text{-name}}(\text{RelResult1})$
 - c) $\text{RelResult3} = \sigma_{e\text{-valor} < 1000 \text{ and } c\text{-cidade} = \text{Rye}}(\text{RelResult2})$

```
SELECT C_NOME  
FROM CLIENTES, EMPRESTIMOS  
WHERE C_NOME = E_NOME AND  
E_VALOR < 1000 AND C_CIDADE = 'RYE'
```

Faça uma lista com todos os números de projetos nos quais esteja envolvido algum empregado cujo último nome seja 'Smith'; ou como empregado, ou como gerente do departamento que controla o projeto.

```
(select distinct pnumero  
from projeto, departamento, empregado  
where Dnum = Dnumero and GerSSN = ssn and  
unome = 'Smith')  
union  
(select distinct pnumero  
from trabalha_em, projeto, empregado  
where pnumero = Pno and essn = SSN and  
unome = 'Smith');
```

Obs: Diferença entre conjuntos → except
Interseção de conjuntos → intersect