DCC060 – Banco de Dados

MATERIAL DE APOIO

Álgebra Relacional

PROF. TARCÍSIO DE SOUZA LIMA

Linguagens Relacionais

- ☐ ***FORMAIS***
  - ***ÁLGEBRA RELACIONAL***
  - ***CÁLCULO RELACIONAL***
    - ***Tuplas***
    - ***Domínios***
- ☐ ***COMERCIAIS***
  - ***SQL***
  - ***QUEL***
  - ***QBE ...***

Operações da Álgebra Relacional

- ☐ A álgebra relacional é composta por um conjunto de operações utilizadas para manipular ***relações*** como um todo.
- ☐ Toda operação relacional é definida sobre uma ou mais relações e o seu resultado ***sempre*** é uma relação, a qual pode ser usada em operações subseqüentes.
- ☐ Do ponto de vista algébrico, uma relação é um elemento imutável, atômico. Assim, não existem operações de inclusão ou modificação de tuplas, ou de definição de relações.
- ☐ Os operadores relacionais são definidos tendo por objetivo atender:
  - **As restrições de uma álgebra**, de maneira a garantir propriedades desejáveis e permitir a preservação (ou o controle) dessas propriedades nas relações resultantes.
  - **As necessidades de implementação**, de maneira a que cada operador corresponda um *algoritmo* que possa ser executado num computador, realizando aquela operação numa base de dados nele armazenada.

Operações da Álgebra Relacional

O conjunto das operações relacionais podem ser divididos em três grupos:

- **Operações sobre Conjuntos**
  - União (∪)
  - Intersecção (∩)
  - Diferença (–)
  - Produto Cartesiano (⊗)
  - Complemento (¬)
  - Complemento Ativo (¬*)
- **Operações Relacionais Unárias**
  - Seleção (σ)
  - Projeção (π)
- **Operações Relacionais Binárias**
  - Junção (⊠)
  - Divisão (÷)

Operações da Álgebra Relacional

- ☐ Além disso a Álgebra Relacional utiliza uma notação (relativamente) padrão, a qual incorpora ainda mais duas ***"operações"***, embora estas não sejam operações no sentido matemático do termo:
  - **Atribuição de Nome a Relações**

    permite atribuir um nome a uma relação que não a tenha - em particular a relações que são o resultado de uma expressão da Álgebra Relacional:

    Nome ⟵ Expressão da Álgebra Relacional
  - **Substituição de Nomes de Atributos**

    permite dar um novo nome para os atributos de uma relação que resulta de uma operação da Álgebra Relacional:

    NomeRel(NomeAtr, ...) ⟵ Expressão da Álgebra Relacional

Por exemplo: *Aluno = { Nome, Idade, Curso }*

Professor = { Nome, Idade, Depto. }

Pessoa ⟵ Aluno $\cup$ Professor *- atribuição de nome*

Pessoa = { Identificação, Idade, Escolaridade } ⟵ Aluno *- substituição de nome*

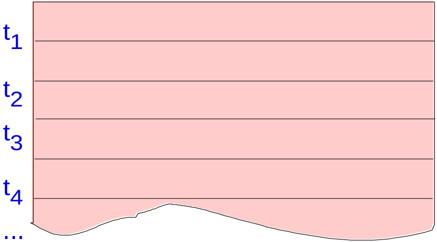
Operadores da Álgebra Relacional

- Operadores para consulta e alteração de relações
- Linguagem procedural
  - uma expressão na álgebra define uma execução seqüencial de operadores
  - a execução de cada operador produz uma relação
- Classificação dos operadores
  - fundamentais
    - unários: *seleção*, *projeção*
    - binários: *produto cartesiano*, *união* e *diferença*
  - derivados
    - binários: *intersecção*, *junção* e *divisão*
  - especiais
    - *renomeação* (unário) e *atribuição*
  - operador de *alteração* (unário)

Operações sobre Conjuntos

- O grupo de operações sobre conjuntos da Álgebra Relacional corresponde às operações usuais da ***Teoria dos Conjuntos***.
- Dentro da Álgebra Relacional elas são definidas considerando-se que cada relação é um conjunto de tuplas:

$R = \{t_1, t_2, t_3, \dots t_n\}$



Para que duas relações possam ser operadas por uma operação sobre conjunto, é necessário que ambas as relações sejam "compatíveis em domínio" (ou "*Union Compatible*")

Operações sobre Conjuntos

☐ Duas relações **$R(A_1, A_2, \ldots A_n)$** e **$S(B_1, B_2, \ldots B_n)$** são ditas **compatíveis em domínio** se ambas têm o mesmo grau **n** e se **$Dom(A_i) = Dom(B_i)$, $1 \leq i \leq n$.** Ou seja, duas relações são ***compatíveis em domínio*** quando além de ter o ***mesmo número de atributos***, cada par de atributos correspondentes têm o ***mesmo domínio***.

Exemplo: *Aluno = { Nome, Idade, Curso }*

Professor = { Nome, Idade, Depto. }

Func. = { Nome, Depto., Idade }

Dom(Nome) = Char(30)
Dom(Idade) = Inteiro
Dom(Curso) = Char(5)
Dom(Depto.) =Char(5)

⇒ ***Aluno*** *é compatível com* ***Professor****, mas não é compatível com* ***Func.***

Note-se que a semântica de uma relação não é importante, mas sim sua estrutura. No caso a ordem dos atributos vale mais do que o fato dos tipos de dados serem distintos.

Operações sobre Conjuntos

☐ As **operações binárias** sobre conjuntos da Álgebra Relacional somente podem operar sobre pares de relações compatíveis em domínio.

Tais operações são as usuais da teoria dos conjuntos:

União: **R ∪ S**	*O resultado contém todas as tuplas de **S**, e todas as tuplas de **R**, porém tuplas que estão em ambas aparecem apenas uma vez*
Intersecção: **R ∩ S**	*O resultado contém apenas as tuplas que estão em **R** e também em **S**.*
Diferença: **R – S**	*O resultado contém as tuplas que estão em **R** mas não estão em **S**.*

☐ Devido às necessidades de consulta, às vezes considera-se uma outra operação:

União Exclusiva: **R ∪| S**	*O resultado contém as tuplas que estão em **S** ou **R**, mas não as tuplas que estão em ambas.*

Esquema Exemplo

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Professor = { Nome, Idade , Depto. }
{ <Ari, 30, Compt.>
<Eva, 27, Eletr.>
<Zeca, 25, Compt.> }

Ministra = { Curso, Depto. }
{ <Compt. , DCC>
<Compt. , MAT>
<Matem., MAT>
<Eletr., DCC>
<Eletr., MAT>
<Eletr., ELE> }

Matrícula = { NomeA, Discipl , Nota }
{ <Zeca, DCC125, 8,5>
<Zico, DCC148, 5,2>
<Juca, DCC125, 6,0>
<Juca, DCC148, 7,0>}

Aulas = { NomeProf, Discipl }
{ <Adão, DCC125>
<Eva, DCC180>
<Adão, DCC148> }

Operações sobre Conjuntos – Exemplos

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Professor = { Nome, Idade , Depto. }
{ <Ari, 30, Compt.>
<Eva, 27, Eletr.>
<Zeca, 25, Compt.> }

União:

Aluno ∪ Professor = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.>
<Ari, 30, Compt.>
<Eva, 27, Eletr.> }

Observação:
Convenciona-se usar os nomes dos atributos da *relação da esquerda*, quando não especificado.

Intersecção:

Aluno ∩ Professor = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.> }

Operações sobre Conjuntos – Exemplos

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Professor = { Nome, Idade , Depto. }
{ <Ari, 30, Compt.>
<Eva, 27, Eletr.>
<Zeca, 25, Compt.> }

Diferença:

Aluno – Professor = { Nome, Idade , Curso }
{ <Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Professor – Aluno = { Nome, Idade , Depto. }
{ <Ari, 30, Compt.>
<Eva, 27, Eletr.> }

Observação:
Nota-se que a operação diferença *não é comutativa*.

Operações sobre Conjuntos – Exemplos

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Professor = { Nome, Idade , Depto. }
{ <Ari, 30, Compt.>
<Eva, 27, Eletr.>
<Zeca, 25, Compt.> }

União Exclusiva:

Aluno ∪| Professor = { Nome, Idade , Curso }
{ <Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.>
<Ari, 30, Compt.>
<Eva, 27, Eletr.> }

Operações sobre Conjuntos

Operação: Complemento de uma Relação ($\neg$ R)

☐ Como as relações são subconjuntos do produto cartesiano dos domínios dos atributos dessa relação, pode-se considerar que o ***universo de uma relação*** é o próprio produto cartesiano.

Seja:

$\mathbf{R = \{A_1, A_2, \ldots A_n\}}$

$\mathbf{Universo(R) = Dom(A_1) \otimes Dom(A_2) \otimes \ldots Dom(A_n)}$

$\neg R = \mathbf{Universo(R) - R}$

$\neg R = \mathbf{Dom(A_1) \otimes Dom(A_2) \otimes \ldots Dom(A_n) - R}$

Observação:
Esta operação até é usada em situações raríssimas, mas usualmente ela não tem muita serventia prática, pois os domínios dos atributos são usualmente conjuntos com muitos elementos

Exemplo:

Aluno = { Nome, Idade , Curso }

{ <Zeca,	25,	Compt.>
<Zico,	18,	Eletr.>
<Juca,	21,	Odonto>
<Tuca,	18,	Compt.> }

Dom(Nome) = Char(30)
Dom(Idade) = Int[0-149]
Dom(Curso) = Char(6)

$Dom(Nome) = 96^{30}$
$Dom(Idade) = 150$
$Dom(Curso) = 96^{6}$

$R_{Aluno} = 4$

$\neg R_{Aluno} = Universo(R_{Aluno}) - R_{Aluno} = 96^{30} * 150 * 96^{6} - 4 = 3.45 * 10^{73} - 4$

Operações sobre Conjuntos

Operação: Complemento Ativo de uma Relação (¬* R)

- Esta operação é definida sobre o conceito de **Domínio Ativo de um Atributo** da Relação.
- Domínio Ativo de um Atributo: Dom*(Atr)(R)

  O Domínio Ativo de um Atributo é definido sobre cada relação que o atributo participa.

  O Domínio Ativo de um Atributo **Atr** definido sobre uma relação **R** é o conjunto de todos os valores que esse atributo assume nessa relação

Exemplo:

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Dom*(Nome)(Aluno) = { Zeca, Zico, Juca, Tuca }
Dom*(Idade)(Aluno) = { 18, 21, 25 }
Dom*(Curso)(Aluno) = { Compt., Eletr., Odonto }

Observação:
Esta operação é muito mais útil em situações práticas.

Operações sobre Conjuntos

Operação: Complemento Ativo de uma Relação (¬* R)

O Complemento Ativo ¬* R , **R = {A_1, A_2, ... A_n}**, é o conjunto:

Dom*(A_1)(R) ⊗ Dom*(A_2)(R) ⊗ ... Dom*(A_n)(R) – R

Continuando o exemplo:

Aluno = { Nome, Idade , Curso }

Dom*(Nome)(Aluno) = {Zeca, Zico, Juca, Tuca} = 4
Dom*(Idade)(Aluno) = {18, 21, 25} = 3
Dom*(Curso)(Aluno) = {Compt., Eletr., Odonto} = 3

¬*Aluno = 4 * 3 * 3 = 36

Operações sobre Conjuntos

Operação: Complemento Ativo de uma Relação (¬* R)

Outro exemplo: Seja a relação que indica quais departamentos participam da ministração de quais cursos

Ministra = { Curso, Depto. }
{ <Compt. , DCC>
<Compt. , MAT>
<Matem., MAT>
<Eletr., DCC>
<Eletr., MAT>
<Eletr., ELE> }

Dom(Curso) = Char(6)
Dom(Depto.) = Char(7)

Dom(Curso) = 96^6
Dom(Depto) = 96^7

¬ Ministra = $96^6 * 96^7 - 6 = 5,8 * 10^{25} - 6$

→ Considere-se sobre esta relação a consulta: *Q1: “Quais são os departamentos que não participam de quais cursos?”*

Resposta: ¬* Ministra

Dom*(Curso)(Ministra) = {Compt., Matem., Eletr. }
Dom*(Depto)(Ministra) = {DCC, MAT, ELE}
¬*Ministra = 3 * 3 – 6 = 3

¬* **Ministra = { Curso, Depto. }**
{<Compt. , ELE>
<Matem., DCC>
<Matem., ELE> }

Operações sobre Conjuntos

Operação: Produto Cartesiano (R ⊗ S)

- O **Produto Cartesiano de duas Relações**, tal como as demais operações sobre conjuntos da Álgebra Relacional, também não leva em conta a estrutura de cada relação. Assim, a operação ***produto cartesiano de duas relações R e S*** tem como resultado uma relação que tem como atributos a concatenação dos atributos da ***relação R*** e da ***relação S***, e tem como tuplas todas as combinações possíveis de ***valores de R*** com ***valores de S***.
- Esta operação difere das demais operações sobre conjuntos no sentido de que, ao contrário das demais, não impõe que as relações devam ser *Compatíveis em Domínio*.

Exemplo:

Ministra = { Curso, Depto. }
{<Compt. , DCC>
<Compt. , MAT>
<Matem., MAT>}

⊗

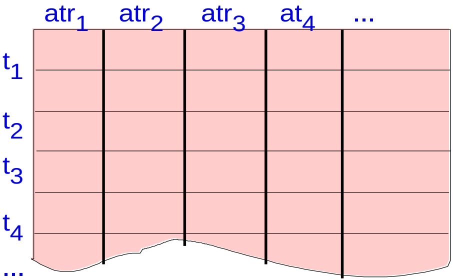
D#Matric = { Depto., Dia }
{<DCC, 25>
<MAT, 26>}

Ministra ⊗ D#Matric =
{ Curso, M.Depto., D.Depto., Dia }
{<Compt. , DCC, DCC, 25>
<Compt. , DCC, MAT, 26>
<Compt. , MAT, DCC, 25>
<Compt. , MAT, MAT, 26>
<Matem., MAT, DCC, 25>
<Matem., MAT, MAT, 26>}

Operações relacionais unárias

- As operações relacionais levam em conta a estrutura interna das Relações, em termos de quais são os atributos que as compõem. Ou seja, não tratam as relações apenas como um conjunto de tuplas, mas como um subconjunto de produtos cartesianos de domínios de atributos:

$R = \{Atr_1, Atr_2, Atr_3, \ldots Atr_n\}$



- Existem basicamente 2 operações relacionais unárias:
  - **Seleção**
  - **Projeção**

Seleção

- Retorna tuplas que satisfazem um predicado
- Resultado
  - subconjunto horizontal de uma relação
- Notação

$$\sigma_{\texttt{predicado}}\ \texttt{(relação)}$$

- Operadores de comparação

$=, <, <=, >, >=, \neq$

- Operadores lógicos: $\wedge$ (and) $\vee$ (or) $\neg$ (not)
- Exemplo: $\sigma_{z\ >=\ 2}(R)$

R

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

resultado

x	*y*	*z*
2	*2*	*2*
2	*2*	*3*

Projeção

- Retorna um ou mais atributos de interesse
- Resultado
  - subconjunto vertical de uma relação
- Notação

$$\pi_{\text{lista\_nomes\_atributos}} \; (\text{relação})$$

- Eliminação automática de 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*

Operações relacionais unárias

Operação: Seleção ($\sigma_{<condição>}$ R)

☐ A **operação de Seleção** aplicada sobre uma relação **R** resulta num sub-conjunto das tuplas de **R** que satisfazem à <condição> indicada.

A <condição> sempre é uma operação de comparação (=, ≤ , > etc.) de um ***atributo da relação*** com uma ***constante*** ou com *outro* ***atributo*** da própria relação, comparando valores de dois atributos da mesma tupla.

Exemplo: Seja a relação de alunos

Aluno = { Nome, Idade , Curso }

{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto> ⇐
<Tuca, 18, Compt.>
<Teca, 19, Compt.>
<Tina, 22, Odonto> } ⇐

A consulta:

Q2: Selecione os alunos que fazem "Odonto"

corresponde à operação: $\sigma_{\text{Curso = "Odonto"}}$ Aluno

o que resulta na relação:

{<Juca, 21, Odonto>
<Tina, 22, Odonto> }

Observação:
Note-se que neste caso a condição compara um atributo da relação com uma constante

Operações relacionais unárias

Operação: Seleção ($\sigma_{<condição>}$ R)

- Existem consultas que comparam dois atributos da mesma relação. Essa forma em geral é aplicada sobre relações que resultam de um Produto Cartesiano:

Exemplo:

Seja a consulta:

Q3: Selecione os Departamentos em que cada aluno cumpre as disciplinas de seu curso.

Aluno = { Nome, Idade , Curso }

Ministra = { Curso, Depto. }

corresponde à seqüência de operações:

AlunoDepto { Nome, Idade, A.Curso, M.Curso, Depto. } ← Aluno ⊗ Ministra

$\sigma_{\underline{A.Curso = M.Curso}}$ AlunoDepto

Observação:
Note-se que ambos os atributos comparados devem pertencer à Relação

A Seleção verifica tupla por tupla se a condição é satisfeita. Sempre que for, aquela tupla passa para a relação resultado. Quando dois atributos são comparados, verifica-se o valor de ambos na mesma tupla.

Operações relacionais unárias

Operação: Projeção ($\pi_{\{atributos\}}$ R)

- A **operação de Projeção** aplicada sobre uma relação **R** resulta numa relação que tem apenas os atributos indicados na lista {atributos}.

A lista {atributos} é um sub-conjunto do conjunto de atributos da própria relação **R**.

Exemplo: Seja a relação de alunos

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.>
<Teca, 19, Compt.>
<Tina, 22, Odonto> }

A consulta:
Q4: Obtenha o nome e a idade dos alunos

corresponde à operação: $\pi_{\{Nome,\ Idade\}}$ Aluno

o que resulta na relação: **{Nome, Idade}**,
contendo as seguintes tuplas:
{ <Zeca, 25>
<Zico, 18>
<Juca, 21>
<Tuca, 18>
<Teca, 19>
<Tina, 22> }

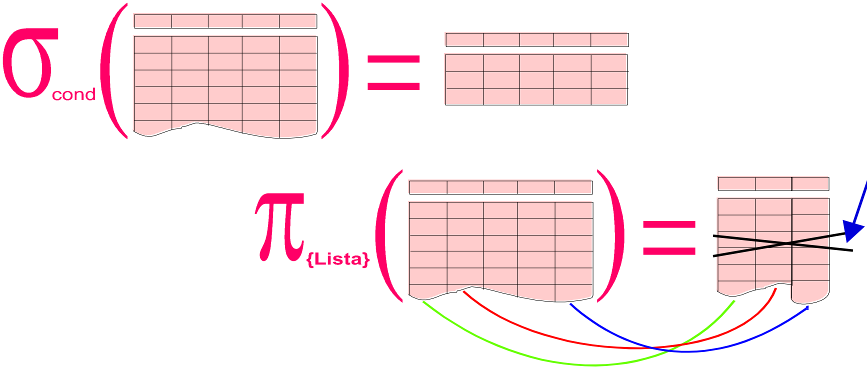
Operações relacionais unárias

Considerações sobre a Operação Projeção

- O resultado de uma operação de projeção é uma relação, o que significa que não devem existir tuplas repetidas no resultado.
- Se a lista de atributos *{atributos}* contiver uma chave da relação, então pode-se ter certeza de que o resultado não terá tuplas repetidas.
- Se a lista de atributos *{atributos}* não contiver uma chave da relação, então poderá haver mais de uma tupla que tenha o mesmo valor para todos os atributos da lista. Nesse caso, tuplas repetidas devem ser eliminadas (Operação de Eliminação de Tuplas Replicadas)

Operações relacionais unárias

- Intuitivamente, a operação de Seleção pode ser vista como escolhendo algumas ***"linhas"*** da tabela que é a relação
- A operação de Projeção pode ser vista como escolhendo algumas ***"colunas"*** da tabela. Neste caso, note-se que algumas tuplas podem ser eliminadas, diminuindo também algumas "linhas" da tabela.



Operações relacionais unárias - Exemplos

Seja a relação de Matrículas
de Alunos em Disciplinas:

Matrícula = { NomeA, Discipl , Nota }
{ <Zeca, DCC125, 8,5>
<Zico, DCC148, 5,2>
<Juca, DCC125, 6,0>
<Juca, DCC148, 7,0>}

Q5: Listar as Notas que os alunos obtiveram na disciplina DCC125

$\pi_{\{\text{NomeA, Nota}\}}(\sigma_{\text{Discip = “DCC125”}}\ \textbf{Matrícula}) =$

$= \pi_{\{\text{NomeA, Nota}\}}$ ({ <Zeca, DCC125, 8,5>
<Juca, DCC125, 6,0>}) = ({ <Zeca, 8,5>
<Juca, 6,0>})

Q6: Listar as Disciplinas em que o aluno “Zeca” está matriculado

$\pi_{\{\text{Discipl}\}}(\sigma_{\text{NomeA = “Zeca”}}\ \textbf{Matrícula}) =$

$= \pi_{\{\text{Discipl}\}}$ ({ <Zeca, DCC125, 8,5>}) = ({ <DCC125>})

Observação: Note-se que embora a resposta seja um único valor, o resultado ainda é uma tupla

Operações relacionais unárias - Exemplos

Sejam as relações de Alunos e Professores

Aluno = { Nome, Idade , Curso }
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Professor = { Nome, Idade , Depto. }
{ <Ari, 30, Compt.>
<Eva, 27, Eletr.>
<Zeca, 25, Compt.> }

Q7: Listar a idade de alunos e professores da base

$\textbf{Aux1} \leftarrow \pi_{\{\text{Nome, Idade}\}} \textbf{Aluno}$

$\textbf{Aux2} \leftarrow \pi_{\{\text{Nome, Idade}\}} \textbf{Professor}$

$\textbf{Pessoas} \leftarrow \textbf{Aux1} \cup \textbf{Aux2}$

Operações relacionais unárias - Exemplos

Sejam as relações de Alunos e suas Matrículas

Aluno = { Nome, Idade , Curso }

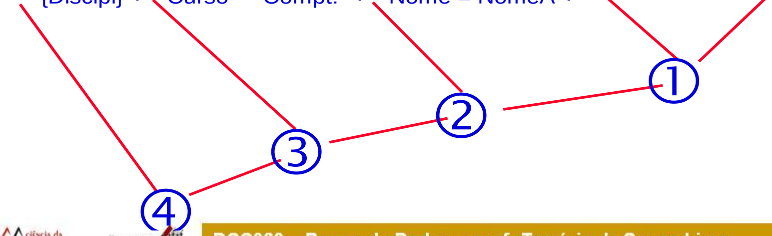
{ <Zeca, 25, Compt.>
<Zico, 18, Eletr.>
<Juca, 21, Odonto>
<Tuca, 18, Compt.> }

Matrícula = { NomeA, Discipl , Nota }

{ <Zeca, DCC125, 8,5>
<Zico, DCC148, 5,2>
<Juca, DCC125, 6,0>
<Juca, DCC148, 7,0>}

Q8: Listar as disciplinas em que alunos de “Compt.” se matricularam

$$\pi_{\{\text{Discipl}\}} (\sigma_{\text{Curso = “Compt.”}} (\sigma_{\text{Nome = NomeA}} (\textbf{Matrícula} \otimes \textbf{Aluno})))$$



Operações relacionais unárias - Exemplos

①

⊗ = { Nome, Idade , Curso, NomeA, Discipl , Nota }
{ <Zeca, 25, Compt., Zeca, DCC125, 8,5>
<Zico, 18, Eletr., Zeca, DCC125, 8,5>
<Juca, 21, Odonto, Zeca, DCC125, 8,5>
<Tuca, 18, Compt., Zeca, DCC125, 8,5>
<Zeca, 25, Compt., Zico, DCC148, 5,2>
<Zico, 18, Eletr., Zico, DCC148, 5,2>
<Juca, 21, Odonto, Zico, DCC148, 5,2>
<Tuca, 18, Compt., Zico, DCC148, 5,2>
<Zeca, 25, Compt., Juca, DCC125, 6,0>
<Zico, 18, Eletr., Juca, DCC125, 6,0>
<Juca, 21, Odonto, Juca, DCC125, 6,0>
<Tuca, 18, Compt., Juca, DCC125, 6,0>
<Zeca, 25, Compt., Juca, DCC148, 7,0>
<Zico, 18, Eletr., Juca, DCC148, 7,0>
<Juca, 21, Odonto, Juca, DCC148, 7,0>
<Tuca, 18, Compt., Juca, DCC148, 7,0> }

Operações relacionais unárias - Exemplos

② $\sigma_{\text{Nome = NomeA}}$

{ <Zeca, 25, Compt., Zeca, DCC125, 8,5> ⇐
<Zico, 18, Eletr., Zeca, DCC125, 8,5>
<Juca, 21, Odonto, Zeca, DCC125, 8,5>
<Tuca, 18, Compt., Zeca, DCC125, 8,5>
<Zeca, 25, Compt., Zico, DCC148, 5,2>
<Zico, 18, Eletr., Zico, DCC148, 5,2> ⇐
<Juca, 21, Odonto, Zico, DCC148, 5,2>
<Tuca, 18, Compt., Zico, DCC148, 5,2>
<Zeca, 25, Compt., Juca, DCC125, 6,0>
<Zico, 18, Eletr., Juca, DCC125, 6,0>
<Juca, 21, Odonto, Juca, DCC125, 6,0> ⇐
<Tuca, 18, Compt., Juca, DCC125, 6,0>
<Zeca, 25, Compt., Juca, DCC148, 7,0>
<Zico, 18, Eletr., Juca, DCC148, 7,0>
<Juca, 21, Odonto, Juca, DCC148, 7,0> ⇐
<Tuca, 18, Compt., Juca, DCC148, 7,0> }

↓

{ <Zeca, 25, Compt., Zeca, DCC125, 8,5>
<Zico, 18, Eletr., Zico, DCC148, 5,2>
<Juca, 21, Odonto, Juca, DCC125, 6,0>
<Juca, 21, Odonto, Juca, DCC148, 7,0>}

Operações relacionais unárias - Exemplos

③ $\sigma_{\text{Curso = “Compt.”}}$

{ <Zeca, 25, Compt., Zeca, DCC125, 8,5>
<Zico, 18, Eletr., Zico, DCC148, 5,2>
<Juca, 21, Odonto, Juca, DCC125, 6,0>
<Juca, 21, Odonto, Juca, DCC148, 7,0>}

{ <Zeca, 25, Compt., Zeca, DCC125, 8,5>}

④ $\pi_{\{\text{Discipl}\}}$

{ <Zeca, 25, Compt., Zeca, DCC125, 8,5>}

{ <DCC125>}

Operações relacionais unárias - Propriedades

- ☐ O operador Seleção é **comutativo**:

$$\sigma_{\text{<Condição-A>}} (\sigma_{\text{<Condição-B>}} \mathbf{R}) = \sigma_{\text{<Condição-B>}} (\sigma_{\text{<Condição-A>}} \mathbf{R})$$

Dessa forma, uma seqüência de Seleções pode ser executada em qualquer ordem, ou pode ser transformada numa única seleção com uma condição conjuntiva (termos cujo valor é VERDADEIRO ou FALSO, ligados pelo operador E (AND)) :

$$\sigma_{\text{<Condição-1>}} (\sigma_{\text{<Condição-2>}} (\ldots(\sigma_{\text{<Condição-n>}} \mathbf{R}))) =$$

$$\sigma_{\text{<Condição-1> } \mathbf{E} \text{ <Condição-2> } \mathbf{E} \text{ <Condição-n>}} \mathbf{R}$$

- ☐ O operador Projeção **não** é comutativo.

Além disso, se *{Atribs_B}* contém *{Atribs_A}* , então ambas as expressões seguintes são correlatas, e vale a igualdade:

$$\pi_{\{\text{Atribs\_A}\}} (\pi_{\{\text{Atribs\_B}\}} \mathbf{R}) = \pi_{\{\text{Atribs\_A}\}} \mathbf{R}$$

Operações relacionais binárias

- Teoricamente, se apenas as seguintes operações forem definidas, todas as demais operações podem ser definidas a partir delas:

  União ($\cup$), Intersecção ($\cap$), Diferença ($-$),

  Produto Cartesiano ($\otimes$), Seleção (σ), Projeção (π)

  Por exemplo, a operação de *intersecção* pode ser definida através apenas da *união* e *diferença*:

  $$\mathbf{R \cap S = (R \cup S) - ((R\text{-}S) \cup (S - R))}$$

  No entanto, por diversas razões é interessante dispor de outras operações, as quais, se não necessárias, são convenientes em determinadas situações.

- Todas as Operações Relacionais Binárias da Álgebra Relacional são possíveis de serem definidas através de produtos cartesianos e outras operações do conjunto básico e, assim, estritamente falando, são desnecessárias.

União, Diferença e Intersecção

- Operam somente sobre duas relações R_1 e R_2 ditas compatíveis
  - grau(R_1) = grau(R_2)
  - para i de 1 até grau(R_1):

    domínio(atributo a_i de R_1) = domínio(atributo a_i de R_2)
- Grau do resultado
  - grau(R_1) (ou grau(R_2))
- Nomes dos atributos do resultado
  - nomes dos atributos da primeira relação (R_1 - relação à esquerda)

União

- Retorna a união das tuplas de de duas relações R_1 e R_2
- Eliminação automática de duplicatas
- Notação

```
relação1  ∪  relação2
```

- Exemplo:

R_1

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

R_2

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

$R_1 \cup R_2$

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

Diferença

- Retorna as tuplas presentes em R_1 e ausentes em R_2
- Notação

  `relação1 — relação2`

- Exemplo:

R_1

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

R_2

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

$R_1 - R_2$

x	y	z
1	2	2
2	2	3

Intersecção

- Retorna as tuplas comuns a R_1 e R_2
- Notação

  `relação1 ∩ relação2`

- Exemplo:

R_1

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

R_2

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

$R_1 \cap R_2$

x	y	z
1	1	1
3	1	1

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
  - cardinalidade(R_1) * cardinalidade(R_2)
- Notação

  `relação1   X   relação2`

- Exemplo:

R_1

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

R_2

w	y
1	1
2	2

$R_1 X R_2$

x	R_1.y	z	w	R_2.y
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

Operações relacionais binárias

- ☐ No entanto, principalmente devido ao fato que podem ser desenvolvidos algoritmos mais eficientes para determinadas operações "compostas" do que seria possível utilizando apenas combinações de algoritmos das operações básicas, consideram-se como fazendo parte da Álgebra Relacional diversas outras operações, chamadas genericamente de ***Operações Relacionais Binárias***.
- ☐ Em geral, após uma operação Produto Cartesiano, é necessário comparar um ou um grupo de atributos de uma das relações originais com um atributo ou grupo de atributos compatíveis em domínio da outra relação.

  Por exemplo, seja a consulta Q8:

  Aluno = { Nome, Idade , Curso } **Matrícula = { NomeA, Discipl , Nota }**

  Q8: Listar as disciplinas em que alunos de "Compt." se matricularam

  Respondida como:

$$\pi_{\{Discipl\}} (\sigma_{Curso = \text{“Compt.”}} (\sigma_{Nome = NomeA} (\mathbf{Matrícula} \otimes \mathbf{Aluno})))$$

② ①

Operações relacionais binárias

É muito mais eficiente um algoritmo que somente grave as tuplas que atendem ao critério de seleção “associado” ao Produto Cartesiano, do que gravar todas as tuplas geradas e depois relê-las uma a uma selecionando aquelas que atendem à condição da seleção

- Por ser muito comum, essa operação foi incluída entre as operações da Álgebra Relacional, sendo denominada **Junção** (“Join”).

$$\pi_{\{Discipl\}}\left(\sigma_{Curso\,=\,\text{“Compt.”}}\left(\overbrace{\sigma_{Nome\,=\,NomeA}\left(\textbf{Matrícula} \otimes \textbf{Aluno}\right)}^{\textbf{Matric} \,\boxtimes_{Nome\,=\,NomeA}\, \textbf{Aluno}}\right)\right)$$

② ①

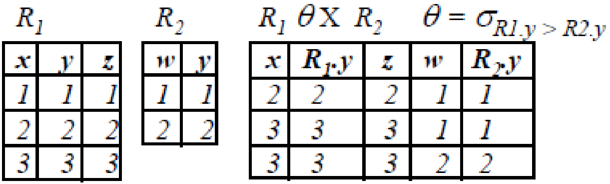
Operações relacionais binárias

Junção (*Join*)

- Retorna a combinação de tuplas de duas relações R_1 e R_2 que satisfazem um predicado
- Notação

  `relação1  θ X  relação2`

- Exemplo:



R_1

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

R_2

w	y
1	1
2	2

$R_1\ \theta X\ R_2 \qquad \theta = \sigma_{R1.y > R2.y}$

x	$R_1.y$	z	w	$R_2.y$
2	2	2	1	1
3	3	3	1	1
3	3	3	2	2

Operações relacionais binárias

Operação Junção $\left(\mathbf{R} \overset{\text{<Condição-J>}}{\boxtimes} \mathbf{S} \right)$

- ☐ A *Condição de Junção <Condição-J>* é da forma:

<Comp_AtrAtr$_1$> E <Comp_AtrAtr$_2$> E ... E <Comp_AtrAtr$_n$>,

sendo <Comp_AtrAtr$_i$> uma comparação entre atributos, ou conjunto de atributos, da forma:

Atrib$_R$ θ Atrib$_S$

Onde Atrib$_R$ é um atributo da relação esquerda R, Atrib$_S$ é um atributo da relação direita S, Atrib$_R$ e Atrib$_S$ têm o mesmo domínio, e θ é uma operação de comparação válida no domínio desses atributos.

$$\pi_{\{\text{Discipl}\}} \underbrace{\left(\sigma_{\text{Curso} = \text{“Compt.”}} \left(\sigma_{\text{Nome} = \text{NomeA}} \left(\textbf{Matrícula} \otimes \textbf{Aluno}\right)\right)\right)}_{\textbf{Matric} \overset{\text{Nome = NomeA E Curso = “Compt.”}}{\boxtimes} \textbf{Aluno}}$$

Operações relacionais binárias

Operação Junção-θ (R $\overset{\text{<Condição-J>}}{\boxtimes}$ S)

- Existem diversas variações sobre a operação de junção, todas elas definidas em razão de sua ampla utilização, e levando-se em conta que assim definidas, cada uma viabiliza uma implementação mais eficiente do que seria possível pela execução das operações elementares que teoricamente a implementa.
- Cada operação de junção tem uma definição própria de como são tratados os atributos envolvidos na comparação e como são tratadas as tuplas onde os atributos envolvidos na comparação têm valor nulo.
- Uma operação de junção denominada **Junção- θ** (ou θ -Join), onde pode ser usado qualquer operador θ válido no domínio dos atributos comparados é uma das mais genéricas.

Na operação Junção-θ , os atributos envolvidos na comparação aparecem ambos na relação resultado, e tuplas com valores nulos nos atributos envolvidos na comparação não aparecem no resultado

Operações relacionais binárias

Operação Equi-Junção ($R \overset{\text{<Condição-J>}}{\boxtimes} S$)

- É comum a operação θ de comparação ser a igualdade, o que é interessante, por simplificar o algoritmo de comparação. Portanto, é interessante dispor de uma Operação de Junção equivalente à Junção- θ que compare *todos os atributos envolvidos com o operador igual (=)*. Essa operação é chamada **Equi-Junção** (ou Equi-Join)

Como na operação Junção-θ , os atributos envolvidos na comparação aparecem ambos na relação resultado, resultando em pares de atributos com valores idênticos na relação resultado. Tuplas com valores nulos nos atributos envolvidos na comparação também não aparecem no resultado.

Existem duas maneiras de simbolizar essa operação:

$$R \overset{\text{<Condição-J>}}{\boxtimes} S \qquad \textbf{OU} \qquad R \overset{\text{\{Atrib\_R\},\{Atrib\_S\}}}{\boxtimes} S$$

Operações relacionais binárias

Operação Junção Natural $\left(R \overset{\{Atrib\_R\},\{Atrib\_S\}}{\bowtie} S \right)$

- Como os atributos comparados numa operação de Equi-Junção aparecem em pares com valores idênticos na relação resultado, um de cada par pode ser eliminado. A operação de **Junção Natural** é semelhante à operação de Equi-Junção, porém, dos atributos comparados (pela operação de igualdade), apenas os originários de uma das relações operadas aparecem na relação resultado.

Na operação Junção-Natural $R \bowtie S$, dos atributos envolvidos na comparação aparecem apenas os originários da relação R na relação resultado. Tuplas com valores nulos nos atributos envolvidos na comparação também não aparecem no resultado.

Assim, é comum considerar-se 3 operações Relacionais Binárias centradas na operação Produto Cartesiano:

$R \overset{<Condição\text{-}J>}{\boxtimes} S$	$R \overset{\{Atrib\_R\},\{Atrib\_S\}}{\boxtimes} S$	$R \overset{\{Atrib\_R\},\{Atrib\_S\}}{\bowtie} S$
Junção-Theta	**Equi-Junção**	**Junção Natural**

Operações relacionais binárias

Junção Natural (*natural join*)

- Junção na qual θ é uma igualdade predefinida entre todos os atributos de mesmo nome presentes em duas relações R_1 e R_2 (atributos de junção). Estes atributos só aparecem uma vez no resultado
- Notação

  `relação1` $\bowtie$ `relação2`
- Derivação

$$R_1 \bowtie R_2 = \pi_{A1,\ ...,\ An,\ B1,\ ...,\ Bm,\ \overbrace{C1,\ ...,\ Cx}^{\text{atributos de junção}}}(R_1\ \theta\ X\ R_2)$$

$$\theta = \sigma_{R1.C1\ =\ R2.C1\ \wedge\ ...\ \wedge\ R1.Cx\ =\ R2.Cx}$$

Operações relacionais binárias
Junção Natural

- Exemplos

R_1

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

R_2

w	y
1	1
2	2

$R_1 \bowtie R_2$

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

R_1

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

R_2

x	y	w
1	1	3
2	2	2

$R_1 \bowtie R_2$

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

Operações relacionais binárias
Junção Natural

- Exemplos

R_1

x	y	z
1	1	1
1	1	2

R_2

w	t
1	1
2	2

$R_1 \bowtie R_2$

x	y	z	w	t
1	1	1	1	1
1	1	1	2	2
1	1	2	1	1
1	1	2	2	2

Operações relacionais binárias

Junções Externas (*outer joins*)

- Junção na qual as tuplas de uma ou ambas as relações que não são combinadas são mesmo assim preservadas no resultado
- Três tipos (exemplos com junção natural)
  - junção externa à esquerda (*left [outer] join*)
    - tuplas da relação à esquerda são preservadas
    - notação: `relação1` ⟕ `relação2`
  - junção externa à direita (*right [outer] join*)
    - tuplas da relação à direita são preservadas
    - notação: `relação1` ⟖ `relação2`
  - junção externa completa (full *[outer] join*)
    - tuplas de ambas as relações são preservadas
    - notação: `relação1` ⟗ `relação2`

Operações relacionais binárias

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	1	1
2	1	2
4	4	4

R_1 ⟕ R_2

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

R_1 ⟖ R_2

x	y	z	a	b
1	1	1	1	1
2	1	2	1	2
4			4	4

R_1 ⟗ R_2

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

Operações relacionais binárias

Operação Divisão (R ÷ S)

☐ Outra operação relacional binária, sem uma ligação intuitiva direta com o produto cartesiano (embora sua definição conceitual o utilize), é a chamada **Divisão**:

$$\mathbf{T \leftarrow R \div S}$$

onde **S** é uma relação cujos atributos são um *subconjunto* dos atributos da relação **R**, ou seja:

$$\mathbf{R(A) \div S(B), \text{ onde } B \subseteq A \text{ e } T(C) \text{ será da forma } C = A - B}$$

☐ Ou seja:

$$\mathbf{T(C) \leftarrow R(C,B) \div S(B)}$$

A Operação Divisão pode ser intuitivamente percebida como uma Divisão Inteira, em que se buscam os registros T(C) cujos valores t_R(C) *ocorrem juntamente com todos os valores* t_S(B), isto é, para cada valor t_R(C) existe uma sub-relação S(B) completa em R(C,B).

Operações relacionais binárias

Divisão

- Considera duas relações
  - dividendo (*grau m* + *n*)
  - divisor (*grau n*)
- *Grau “n”*
  - atributos de mesmo nome em ambas as relações
- Quociente
  - grau “m”
  - atributos da relação dividendo cujos valores associam-se com todos os valores da relação divisor
- Notação

`relação1 ÷ relação2`

Operações relacionais binárias

Divisão

- Exemplos

R_1

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

R_2a

z
1

R_2b

y	z
1	1

R_2c

y
1
2

$R_1 \div R_2a$

x	y
1	1
1	2
2	1

$R_1 \div R_2b$

x
1
2

$R_1 \div R_2c$

x	z
1	1

Operações relacionais binárias

Operação Divisão (R ÷ S)

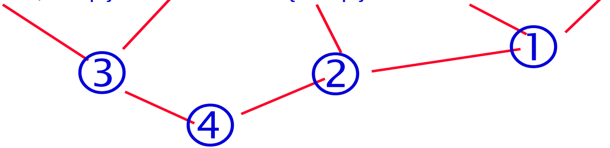
Por exemplo, sejam as relações:

Matrícula = { NomeA, Discipl , Nota }
{ <Zeca, DCC125, 8,5>
<Zico, DCC148, 5,2>
<Juca, DCC125, 6,0>
<Juca, DCC148, 7,0>}

Aulas = { NomeProf, Discipl }
{ <Adão, DCC125>
<Eva, DCC180>
<Adão, DCC148> }

*Q9: Quais alunos cursam **todas** as disciplinas ministradas pelo Prof. Adão?*

R9: $\pi_{\{NomeA,Discipl\}}\ \textbf{Matrícula} \div \pi_{\{Discipl\}}\ \sigma_{NomeProf\ =\ 'Adão'}\ \textbf{Aulas}$



Operações relacionais binárias

Operação Divisão (R ÷ S)

Aulas = { NomeProf, Discipl }
{ <Adão, DCC125>
<Eva, DCC180>
<Adão, DCC148> }

① $\sigma_{\text{NomeProf = 'Adão'}}$ **Aulas =**
{ <Adão, DCC125>
<Adão, DCC148> }

② $\pi_{\{\text{Discipl}\}}\ \sigma_{\text{NomeProf = 'Adão'}}$ **Aulas =**
{ <DCC125>
<DCC148> }

③ $\pi_{\{\text{NomeA,Discipl}\}}$ **Matrícula =**
{ <Zeca, DCC125>
<Zico, DCC148>
<Juca, DCC125>
<Juca, DCC148>}

Matrícula = { NomeA, Discipl , Nota }
{ <Zeca, DCC125, 8,5>
<Zico, DCC148, 5,2>
<Juca, DCC125, 6,0>
<Juca, DCC148, 7,0>}

Operações relacionais binárias

Operação Divisão (R ÷ S)

④ $\pi_{\{NomeA,Discipl\}}$ **Matrícula** ÷ $\pi_{\{Discipl\}}\ \sigma_{NomeProf\ =\ 'Adão'}$ **Aulas**

{ <Zeca, DCC125>
<Zico, DCC148>
<Juca, DCC125>
<Juca, DCC148>}
÷
{ <DCC125>
<DCC148> }
=
{ <Zeca> ?
<Zico> ?
<Juca>} ?

{ <Zeca, DCC125>}
÷
{ <DCC125>
<DCC148> }

Não !

{<Zico, DCC148>}
÷
{ <DCC125>
<DCC148> }

Não !

{<Juca, DCC125>
<Juca, DCC148>}
÷
{ <DCC125>
<DCC148> }

OK

R9: $\pi_{\{NomeA,Discipl\}}$ **Matrícula** ÷ $\pi_{\{Discipl\}}\ \sigma_{NomeProf\ =\ 'Adão'}$ **Aulas =**
{<Juca>}

Esquema Relacional Exemplo

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)

Atribuição

- Armazena o resultado de uma expressão algébrica em uma variável de relação
  - permite o processamento de uma consulta por etapas
- Notação

  nomeVariável ← expressãoÁlgebra
- Exemplo (exercício 1 de produto cartesiano)

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

  $R2 \leftarrow \pi_{codm,\ nome}(Médicos)$

  $Resposta \leftarrow \pi_{nome,\ data}(\sigma_{R1.codm\ =\ R2.codm}(R1\ X\ R2))$

Otimização Algébrica

- Antecipação de seleções
  - filtragens horizontais o mais cedo possível
- Definição de projeções
  - filtragens verticais o mais cedo possível
    - desde que não prejudiquem operações algébricas futuras que necessitem de atributos eliminados
- Identificação de sub-expressões comuns
  - processá-la uma única vez, mantendo-a em uma variável de relação
  - esta variável de relação é usada várias vezes no processamento da consulta

Exemplo de Otimização

- Buscar o nome dos médicos que estão internados como pacientes, sofrendo de hepatite

$$\pi_{\text{Médicos.nome}}\left(\sigma_{\text{Pacientes.CPF = Médicos.CPF} \wedge \text{doença = 'hepatite'}}(\text{Pacientes X Médicos})\right)$$

antecipando seleções e definindo projeções

$$\pi_{\text{nome}}\left(\sigma_{\text{Pacientes.CPF = Médicos.CPF}}\left(\pi_{\text{CPF}}(\sigma_{\text{doença = 'hepatite'}}(\text{Pacientes}))) \text{ X } (\pi_{\text{CPF, nome}}(\text{Médicos}))\right)\right)$$

Renomeação

- Altera o nome de uma relação e/ou dos seus atributos
- Notação

$\rho_{\texttt{(nome\_atributo1, ..., nome\_atributoN)}\ \text{E/OU}\ \texttt{nome\_relação}}\texttt{(relação)}$

- Exemplos

R

x	y	z
1	1	1
2	1	3

$R \times \rho_{R1}(R)$

$R.x$	$R.y$	$R.z$	$R1.x$	$R1.y$	$R1.z$
1	1	1	1	1	1
1	1	1	2	1	3
2	1	3	1	1	1
2	1	3	2	1	3

$\rho_{(a,\ b,\ c)}(R)$

a	b	c
1	1	1
2	1	3

Exemplo de Otimização

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

$$\pi_{\text{M.nroa}}(\sigma_{\text{Médicos.nroa = M.nroa} \wedge \text{Médicos.codm} \neq \text{M.codm}}((\sigma_{\text{cidade = 'Fpolis'}}(\text{Médicos}))\ \text{X}\ (\rho_{\text{M}}(\sigma_{\text{cidade = 'Fpolis'}}(\text{Médicos})))))$$

definindo projeções e identificando sub-expressões em comum

$$\text{R1} \leftarrow \pi_{\text{codm, nroa}}(\sigma_{\text{cidade = 'Fpolis'}}(\text{Médicos}))$$

$$\text{Resposta} \leftarrow \pi_{\text{R1.nroa}}(\sigma_{\text{R1.nroa = R2.nroa} \wedge \text{R1.codm} \neq \text{R2.codm}}(\text{R1}\ \text{X}\ \rho_{\text{R2}}(\text{R1})))$$

Atualização de Relações

- Exclusão
  - notação
    - $\text{relação} \leftarrow \text{relação} - \textit{expressãoConsulta}$
    - $\text{relação} \leftarrow \textit{expressãoConsulta}$
    - $\textit{expressãoConsulta}$ envolve relação
- Inclusão
  - notação
    - $\text{relação} \leftarrow \text{relação} \cup \textit{Expr}$
    - $\textit{Expr}$: conjunto de tuplas
- Alteração
  - notação
    - $\delta_{\{\text{nome\_atributo} \leftarrow \textit{Expr}\}}(\text{relação})$
    - $\textit{Expr}$: expressão aritmética ou 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

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

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

2)

a) $R_1 \leftarrow R_1 \cup \{(1,2,2),\ (1,2,3)\}$

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

$R_1 \leftarrow R_1 \cup (temp\ X\ \{(3,3)\})$

3)

a) $\delta_{x \leftarrow x+1}(R_1)$

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

$R_2 \leftarrow R_2 - temp$

$\delta_{w \leftarrow w-1}(temp)$

$R_2 \leftarrow R_2 \cup temp$

Álgebra Relacional - Exercícios

Seja a Base de Dados:

Fornecedor = { #Forn, FNome, Status , Cidade }
{ <F1, Smith, 20, Londres>
<F2, Junot, 10, Paris>
<F3, Blaise, 30, Paris>
<F4, Clark, 20, Londres>
<F5, Greco, 30, Atenas> }

Peça = { #Peça, PNome, Cor, Peso, Cidade }
{ <P1, Parafuso, vermelho, 12, Londres>
<P2, Pino, verde, 17, Paris>
<P3, Porca, azul, 17, Roma>
<P4, Eixo, vermelho, 14, Londres>
<P5, Parafuso, azul, 12, Paris>
<P6, Porca, vermelho, 19, Londres> }

Fornecimento =
{ #Forn, #Peça, Quant.}
{ <F1, P1, 300>
<F1, P2, 200>
<F1, P3, 400>
<F1, P4, 200>
<F1, P5, 100>
<F1, P6, 100>
<F2, P1, 300>
<F2, P2, 400>
<F3, P2, 200>
<F4, P2, 200>
<F4, P4, 300>
<F4, P5, 400> }

Álgebra Relacional - Exercícios

*Q1: Quais são os **fornecedores** situados em **Londres**?*

R1: $\sigma_{Cidade = \text{'Londres'}}$ **Fornecedor =**

{ <F1, Smith, 20, Londres>
<F4, Clark, 20, Londres> }

*Q2: Quais são os **nomes dos fornecedores** situados em **Londres**?*

R2: $\pi_{\{FNome\}} \sigma_{Cidade = \text{'Londres'}}$ **Fornecedor =**

{ <Smith>
<Clark> }

Fornecedor = { #Forn, FNome, Status , Cidade }

#Forn	FNome	Status	Cidade
{ <F1,	Smith,	20,	Londres>
<F2,	Junot,	10,	Paris>
<F3,	Blaise,	30,	Paris>
<F4,	Clark,	20,	Londres>
<F5,	Greco,	30,	Atenas> }

Álgebra Relacional - Exercícios

*Q3: Quais são os **nomes** dos **fornecedores** que **fornecem a peça P2**?*

R3: $\pi_{\{FNome\}}\,(\underbrace{\sigma_{\#Peça\,=\,'P2'}(\sigma_{F\#Forn\,=\,FP\#Forn}\,(\mathbf{Fornecedor} \otimes \mathbf{Fornecimento})}\,)\,)$

OU $\pi_{\{FNome\}}\,(\,\mathbf{Fornecedor} \overset{\#Peça\,=\,'P2'}{\bowtie} \mathbf{Fornecimento})\,=$

{ < Smith >
< Junot >
< Blaise >
< Clark > }

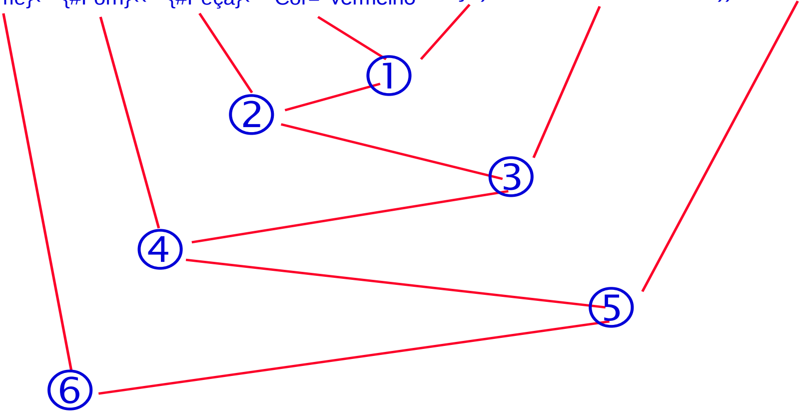
F ☞ FP = {#Forn, FNome, Status , Cidade, #Peça, Quant.}

{<F1,	Smith,	20,	Londres,	P1,	300>
<F1,	Smith,	20,	Londres,	P2,	200>
<F1,	Smith,	20,	Londres,	P3,	400>
<F1,	Smith,	20,	Londres,	P4,	200>
<F1,	Smith,	20,	Londres,	P5,	100>
<F1,	Smith,	20,	Londres,	P6,	100>
< F2,	Junot,	10,	Paris,	P1,	300>
< F2,	Junot,	10,	Paris,	P2,	400>
< F3,	Blaise,	30,	Paris,	P2,	200>
< F4,	Clark,	20,	Londres,	P2,	200>
< F4,	Clark,	20,	Londres,	P4,	300>
< F4,	Clark,	20,	Londres,	P5,	400> }

Álgebra Relacional - Exercícios

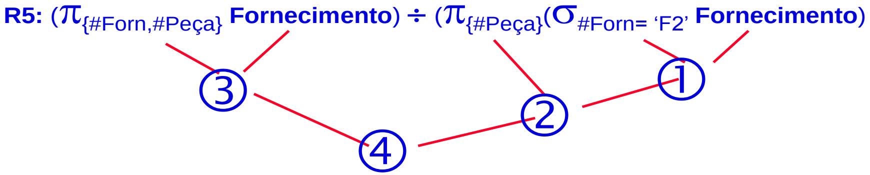
Q4: Quais são os ***nomes*** *dos* ***fornecedores*** *que fornecem <u>ao menos</u>* ***<u>uma</u> peça vermelha****?*

R4: $\pi_{\{Fnome\}}(\pi_{\{\#Forn\}}((\pi_{\{\#Peça\}}(\sigma_{Cor= \text{'vermelho'}}\ \textbf{Peça}) \bowtie \textbf{Fornecimento})) \bowtie \textbf{Fornecedor})$



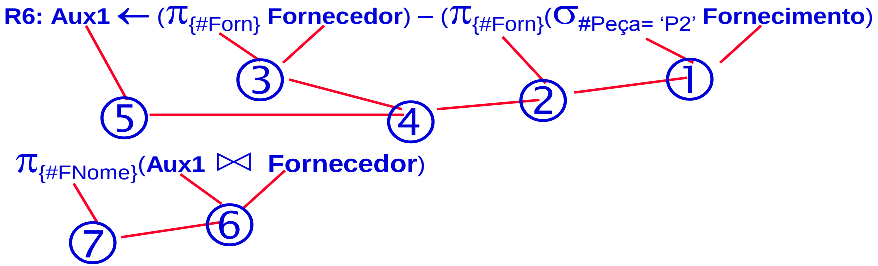
Álgebra Relacional - Exercícios

*Q5: Quais são os **números** dos **fornecedores** que fornecem ao menos **todas as peças fornecidas pelo fornecedor F2**?*



R5: $(\pi_{\{\#Forn,\#Peça\}}\ \textbf{Fornecimento}) \div (\pi_{\{\#Peça\}}(\sigma_{\#Forn= 'F2'}\ \textbf{Fornecimento})$

*Q6: Quais são os **nomes** dos **fornecedores** que não fornecem a **peça P2**?*



R6: $\textbf{Aux1} \leftarrow (\pi_{\{\#Forn\}}\ \textbf{Fornecedor}) - (\pi_{\{\#Forn\}}(\sigma_{\#Peça= 'P2'}\ \textbf{Fornecimento})$

$\pi_{\{\#FNome\}}(\textbf{Aux1} \bowtie \textbf{Fornecedor})$

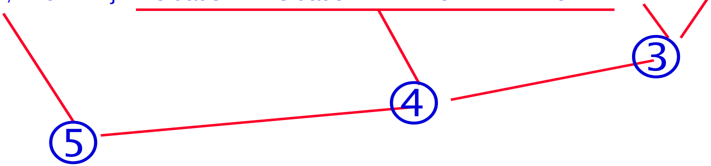
Álgebra Relacional - Exercícios

*Q7: Quais são <u>todos os pares</u> de **números** de **fornecedores** tal que dois fornecedores são localizados na <u>mesma</u> **cidade**?*

R7: $\mathbf{A1} \leftarrow (\pi_{\{\#Forn,\ Cidade\}}\ \mathbf{Fornecedor})$ ①

$\mathbf{A2} \leftarrow (\pi_{\{\#Forn,\ Cidade\}}\ \mathbf{Fornecedor})$ ②

$(\pi_{\{\#Forn.A1,\ \#Forn.A2\}}(\sigma_{Cidade.A1=\ Cidade.\ A2\ E\ \#Forn.A1\ <\ \#Forn.A2}(\mathbf{A1} \otimes \mathbf{A2})$



R7 = { #Forn.A1, #Forn.A2}
{ <F1, F4>
<F2, F3>}