Bases de Dados
–Uma Recordação da Algebra Relacional–

Caetano Traina Jr.

Grupo de Bases de Dados e Imagens
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos
caetano@icmc.usp.br

7 de março de 2013
São Carlos, SP - Brasil

Esta apresentação faz uma recordação da álgebra relacional, revisitando seus principais operadores e aqueles que estendem a álgebra tal como ela é usada nos SGBDR modernos.

Outline

Introdução

- A álgebra relacional é composta por um conjunto de operadores, utilizados para manipular Relações como um todo.
- Todo Operador Relacional é definido sobre uma ou mais relações, e seu resultado sempre é uma relação, a qual pode ser utilizada por operadores subsequentes.
- Do ponto de vista algébrico, uma relação é um elemento imutável, atômico. Assim, não existem operadores de inclusão ou modificação de tuplas, ou de definição de relações.

Introdução

- 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 — Propriedades.
  - **As necessidades de Implementação**: de maneira a que cada operador corresponda a um **algoritmo** que possa ser executado num computador, realizando a operação sobre uma base de dados nele armazenado — Custo.

Operadores Relacionais

Os operadores relacionais podem ser divididos em 3 grupos:

- Operadores sobre Conjuntos
  - União $\bigcup$
  - Intersecção $\bigcap$
  - Diferença $-$
  - União exclusiva $\bigcup|$
  - Complemento $\neg$
  - Complemento Ativo $\neg^*$
  - Produto Cartesiano $\times$
- Operadores Relacionais Unários
  - Seleção σ
  - Projeção π
- Operadores Relacionais Binários
  - Junção natural $*$
  - equijoin $\overset{=}{\bowtie}$
  - junção-θ $\overset{\theta}{\bowtie}$

Renomeação/Atribuição de Nome a Relações e a Atributos

Além dos Operadores Relacionais, utiliza-se ainda o operador "Renomear" simbolizado por ρ, que atende a duas necessidades:

- Atribuição de Nome a Relações – $\boxed{\rho(R_{old} \backslash R_{new})}$
- Substituição de Nomes de Atributos – $\boxed{\rho_{\{a_{old} \backslash a_{new}\}}(R_{old} \backslash R_{new})}$

Renomeação/Atribuição de Nome a Relações e a Atributos

Exemplo

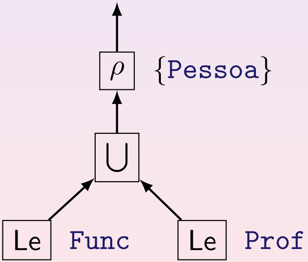
Por exemplo, dado que:

```
Funcionario={Nome, Idade, Depto}
  Professor={Nome, Idade, Depto}
```

- Atribuição de Nome a Relações:

  $\rho\,((Funcionario \bigcup Professor) \backslash Pessoa)$

(A relação de entrada é a que tem o nome trocado)



Renomeação/Atribuição de Nome a Relações e a Atributos

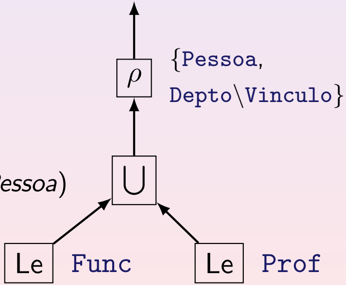
Exemplo

Por exemplo, dado que:

```
Funcionario={Nome, Idade, Depto}
  Professor={Nome, Idade, Depto}
```

- Substituição de Nomes de Atributos

$$\rho_{\{Nome, Idade, Depto \backslash Vinculo\}}((Aluno \bigcup Professor) \backslash Pessoa)$$



Renomeação/Atribuição de Nome a Relações e a Atributos

Exemplo

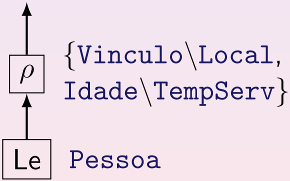
Por exemplo, dado que:

```
Funcionario={Nome, Idade, Depto}
  Professor={Nome, Idade, Depto}
```

- Substituição de Nomes de Atributos

  $\rho_{\{Vinculo \backslash Local, Idade \backslash TempoDeServico\}}(Pessoa)$

(Renomear só atributos)



Renomeação/Atribuição de Nome a Relações e a Atributos

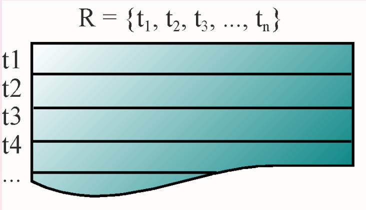
Propriedades do operador Renomear — ρ

Propriedades do operador Renomear:

- O operador de Renomear atributos é idempotente
  $\rho_{\{a \backslash b\}}(\rho_{\{a \backslash b\}} R) \Leftrightarrow \rho_{\{a \backslash b\}} R$
- O operador de Renomear atributos é comutativo
  $\rho_{\{a \backslash b\}}(\rho_{\{c \backslash d\}} R) \Leftrightarrow \rho_{\{c \backslash d\}}(\rho_{\{a \backslash b\}} R) R$
  (desde que a, b, c e d não tenham nomes de atributos em comum)

Operadores sobre Conjuntos

- O grupo dos Operadores sobre Conjuntos da Álgebra Relacional corresponde àqueles bem conhecidos da Teoria dos Conjuntos.
- Para que duas relações possam ser operadas por um operador sobre conjunto, é necessário que ambas sejam "Compatíveis em Domínio".
- Dentro da Álgebra Relacional, elas são definidas considerando-se que cada relação é um conjunto de tuplas.



Operadores sobre Conjuntos

Operadores Unários

- Complemento de uma relação – $\boxed{\neg R}$
  - Na Álgebra Relacional, uma Relação é definida como "um subconjunto do produto cartesiano dos domínios dos atributos".
  - Portanto, o universo de uma relação é o próprio produto cartesiano dos domínios dos atributos. Portanto $\boxed{\neg R\text{=Universo (R)-R}}$.
  - O operador Complemento não é usado em aplicações reais, mas é necessária para a definição da álgebra.
- Complemento ativo de uma relação – $\boxed{\neg^* R}$
  - o operador Complemento Ativo foi criado por ser mais útil na prática.
  - Ele é definido sobre o Domínio Ativo de um Atributo da Relação:
  - O Domínio Ativo $\boxed{Dom^*(A)[R]}$ de um Atributo A numa Relação R é o conjunto de todos os valores que o atributo assume na relação.
  - Portanto o Universo Ativo são as tuplas formadas por todas as combinações possíveis de valores que cada atributo assume na relação, e o complemento ativo são essas tuplas que não estão em R.

Operadores sobre Conjuntos

Operadores binários

- Os operadores relacionais são os usuais da teoria dos conjuntos:
  - União: – $\boxed{R_1 \bigcup R_2}$
    O resultado contém todas as tuplas de R_1 e todas as tuplas de R_2, porém tuplas que estão em ambas as relações aparecem apenas uma vez.
  - Intersecção – $\boxed{R_1 \bigcap R_2}$
    O resultado contém apenas as tuplas que estão em R_1 e também em R_2.
  - Diferença – $\boxed{R_1 - R_2}$
    O resultado contém as tuplas que estão em R_1 mas não estão em R_2.
- Alguns autores também consideram este outro operador:
  - União Exclusiva – $\boxed{R_1 \bigcup | R_2}$
    O resultado contém todas as tuplas que estão nas relações R_1 ou R_2, mas não as tuplas que estão em ambas R_1 e R_2.

Operadores sobre Conjuntos

Propriedades dos Operadores sobre Conjuntos

Propriedades dos operadores $\boxed{\bigcup}$, $\boxed{\bigcap}$ e $\boxed{-}$:

- Os operadores de União, Intersecção e União exclusiva são comutativos:
  $R_1 \bigcup R_2 \Leftrightarrow R_2 \bigcup R_1$
  $R_1 \bigcap R_2 \Leftrightarrow R_2 \bigcap R_1$
  $R_1 \bigcup| R_2 \Leftrightarrow R_2 \bigcup| R_1$
  Note-se que $R_1 - R_2 \neq R_2 - R_1$
- Os operadores de União, Intersecção e União exclusiva são associativos:
  $(R_1 \bigcup R_2) \bigcup R_3 \Leftrightarrow R_1 \bigcup (R_2 \bigcup R_3)$
  $(R_1 \bigcap R_2) \bigcap R_3 \Leftrightarrow R_1 \bigcap (R_2 \bigcap R_3)$
  $(R_1 \bigcup| R_2) \bigcup| R_3 \Leftrightarrow R_1 \bigcup| (R_2 \bigcup| R_3)$
  Note-se que $(R_1 - R_2) - R_3 \neq R_1 - (R_2 - R_3)$
- Os operadores de União e Intersecção são idempotentes:
  $R_1 \bigcup R_1 \Leftrightarrow R_1$
  $R_1 \bigcap R_1 \Leftrightarrow R_1$
  Mas $R_1 - R_1 = \emptyset$ e $R_1 \bigcup| R_1 = \emptyset$

Operadores sobre Conjuntos

Produto Cartesiano

Produto Cartesiano – $\boxed{R_1 \times R_2}$

- O operador da álgebra relacional "Produto Cartesiano", tal como os demais operador sobre conjuntos, também não leva em conta a estrutura das relações.
- Mas o Produto Cartesiano difere dos demais, no sentido de que, ao contrário daqueles, este não impõe que as relações devem ser Compatíveis de Domínio.
- O operador Produto Cartesiano das Relações R_1 e R_2 tem como resultado outra relação em que:
  - Os atributos são a concatenação dos atributos da relação R_1 e da relação R_2,
  - as tuplas são todas as combinações possíveis de valores de R_1 com valores de R_2.

Operadores sobre Conjuntos

Propriedades do operador Produto Cartesiano — $\times$

Propriedades do operador Produto Cartesiano:

- Do ponto de vista da Álgebra Relacional, o operador de Produto Cartesiano é comutativo. Isto é, assumindo que $R_1 = \{A\}$ e $R_2 = \{B\}$ então $R_1 \times R_2 = \{A\} \cup \{B\}$ e portanto vale a propriedade comutativa:
  $R_1 \times R_2 \Leftrightarrow R_2 \times R_1$

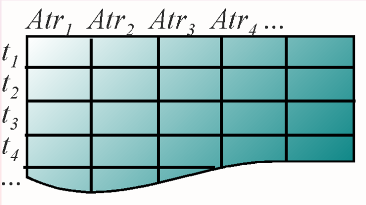
  (Note-se que pela Teoria dos conjuntos, o resultado de um Produto Cartesiano é um par ordenado, e portanto ele não é comutativo.)
- O operador de Produto Cartesiano é associativo:
  $(R_1 \times R_2) \times R_3 \Leftrightarrow R_1 \times (R_2 \times R_3)$
- O operador de Produto Cartesiano não é idempotente:
  $R_1 \times R_1 \neq R_1$

Operadores Relacionais Unárias

- Os operadores relacionais levam em conta a estrutura interna das relações, reconhecendo os atributos que as compõem.
- Portanto, tratam as relações como um subconjunto de produtos cartesianos de domínios de atributos:

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

- Existem basicamente 2 operadores relacionais unários:
  - Seleção – $\boxed{\sigma_{(<condicão>)} R}$
  - Projeção – $\boxed{\pi_{\{<lista\ de\ atributos>\}} R}$



Operadores Relacionais Unários

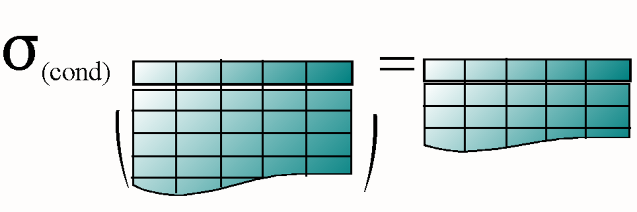
Operador de Seleção

- Operador de Seleção – $\boxed{\sigma_{(c)} R}$
  - Aplicada sobre uma relação R, o operadorde seleção obtém o subconjunto das tuplas que satisfazem à condição c.
  - A condição c sempre é uma operação de comparação θ de um atributo a_i da relação R com:
    - Uma constante: $c := R.a_i \theta cte$;
    - Ou com outro atributo da própria relação, sempre comparando os valores de dois atributos da mesma tupla: $c := R.a_i \theta R.a_j$.
  - O operador de comparação θ é qualquer operador válido no domínio do atributo $R.a_i$.

Operadores Relacionais Unários

Operador de Seleção

- Intuitivamente, a operação de Seleção pode ser vista como a escolha de algumas "linhas" da tabela que armazena a relação.



Operadores Relacionais Unários

Propriedades do operador de seleção — σ

Propriedades do operador de seleção:

- O operador de Seleção é comutativo
  $\sigma_{(<condicão_1>)}\left(\sigma_{(<condicão_2>)}R\right) \Leftrightarrow \sigma_{(<condicão_2>)}\left(\sigma_{(<condicão_1>)}R\right)$
- Dessa forma, uma seqüência de operadores de seleção pode ser executada em qualquer ordem.
- ela também pode ser transformada numa única seleção com uma condição conjuntiva (termos cujo valor é `VERDADEIRO` ou `FALSO`, ligados pelo operador $\wedge$ `(E, AND)`):
  $\sigma_{(<condicão_1>)}\left(\sigma_{(<condicão_2>)}\left(\ldots\left(\sigma_{(<condicão_n>)}R\right)\right)\right) \Leftrightarrow$
  $\sigma_{(<condicão_1>\wedge<condicão_2>\wedge\ldots\wedge<condicão_n>)}R$

Operadores Relacionais Unários

Propriedades do operador de seleção — σ

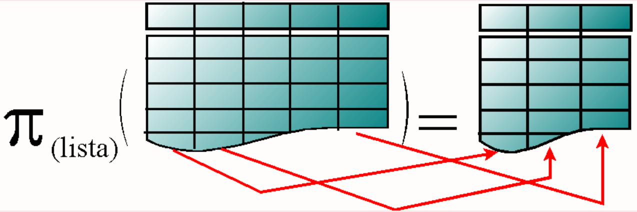
Outras propriedades:

- $\sigma_{(<condição_1>\wedge<condição_2>)}R \Leftrightarrow \sigma_{(<condição_1>)}R \cap \sigma_{(<condição_2>)}R$
- $\sigma_{(<condição_1>\vee<condição_2>)}R \Leftrightarrow \sigma_{(<condição_1>)}R \cup \sigma_{(<condição_2>)}R$
- Idempotência:
  $\sigma_{(<condição_1>)}\left(\sigma_{(<condição_1>)}R\right) \Leftrightarrow \sigma_{(<condição_1>)}R$
- $\sigma_{(\neg<condição_1>)}R \Leftrightarrow R - \sigma_{(<condição_1>)}R$
- $\sigma_{(\neg*<condição_1>)}R \Leftrightarrow R - \sigma_{(<condição_1>)}R$

Operadores Relacionais Unários

Operador de Projeção

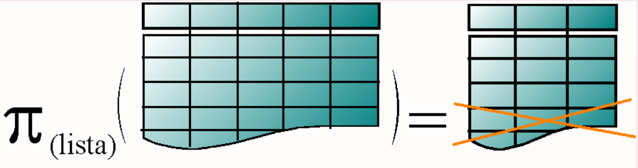
- Operador de Projeção – $\boxed{\pi_{\{<\mathit{lista\ de\ atributos}>\}} R}$
    - o operador de projeção aplicada sobre uma relação R tem como resultado outra relação que tem apenas os atributos indicados na $<$*lista de atributos*$>$.
    - A $<$*lista de atributos*$>$ é um subconjunto dos atributos da própria relação.
- Intuitivamente, a operação de Projeção pode ser vista como a escolha de algumas "Colunas" da tabela.



Operadores Relacionais Unários

Operador de Projeção

- Se a <*lista de atributos*> contiver uma chave da relação, então o resultado não terá tuplas repetidas;
- Se a <*lista de atributos*> não contiver uma chave, então poderá haver mais de uma tupla que tenha o mesmo valor para todos os atributos da lista;
- portanto um operador de Projeção, além de descartar algumas "colunas" da tabela, pode descartar também algumas tuplas repetidas.



Operadores Relacionais Unários

Propriedades do operador de projeção — π

Propriedades do operador de projeção:

- Operador de Projeção não é comutativo.
- Se o subconjunto de atributos B contém o subconjunto A, então vale a igualdade: $\pi_{\{A\}}\left(\pi_{\{B\}} R\right) \Leftrightarrow \pi_{\{A\}} R$

Operadores Relacionais Unários

Distributividade entre Operadores Unários — π e σ

Distributividade entre Operadores de Seleção e Projeção — π e σ:

- Dado dois conjuntos de atributos A e B tal que $A \subseteq B$ então: $\sigma_{(A)}\left(\pi_{\{B\}}R\right) \Leftrightarrow \pi_{\{B\}}\left(\sigma_{(A)}R\right)$

Operadores Relacionais Binários

- Pela Teoria da Álgebra Relacional, se estiverem definidas apenas os seguintes operadores, todos os demais podem ser definidos a partir destes:
  - União $\bigcup$
  - Diferença $-$
  - Produto Cartesiano $\times$
  - Seleção σ
  - Projeção π
  - Renomear relações e atributos ρ
- Esse é o conjunto das "operadores independentes"
- Por exemplo, o Operador de Intersecção é dito ser dependente, porque pode ser definido usando apenas a diferença: $R_1 \bigcap R_2 \Leftrightarrow R_1 - (R_1 - R_2)$

Operadores Relacionais Binários

- Portanto, todos os operador relacionais binários da álgebra relacional são passíveis de serem definidos através de produto cartesiano e outros operadores do conjunto independente. Assim, estritamente falando, todos são desnecessários.
- Como podem ser desenvolvidos algoritmos mais eficientes para determinados operadores dependentes do que seria possível apenas combinando os operadores independentes, consideram-se como fazendo parte da álgebra relacional diversos outros operadores, dos quais os mais importantes são aqueles chamados genericamente de Operadores Relacionais Binários.
- Isso inclui:
  - a Junção natural — $\boxed{R_1 * R_2}$
  - a equijunção — $\boxed{R_1 \overset{a_1=a_2}{\bowtie} R_2}$
  - e a junção-θ — $\boxed{R_1 \overset{a_1 \theta a_2}{\bowtie} R_2}$

Operadores Relacionais Binários

Junção-θ

- Operador de Junção-θ — $\boxed{R_1 \overset{c}{\bowtie} R_2}$
- A condição de Junção c segue a forma:
  $c :=< R_1.a_1\theta_1 R_2.a_1 > \wedge < R_1.a_2\theta_2 R_2.a_2 > \wedge \ldots \wedge < R_1.a_k\theta_k R_2.a_k >$,
  onde:
  - $R_1.a_i$ e $R_2.a_i$ têm o mesmo domínio, e
  - θ_i é uma operação de comparação válida nesse domínio.
- A relação resultado tem todos os atributos de ambas as relações.

Operadores Relacionais Binários

Equijunção

- Operador de Equijunção — $\boxed{R_1 \bowtie_{R_1.a_1=R_2.a_1} R_2}$ ou $\boxed{R_1 \bowtie_{R_1.a_1,R_2.a_1} R_2}$
- O Operador de Equijunção é um caso particular da junção-teta, quando θ é "$=$".
- O motivo dos dois serem tratados de maneira distinta é porque a comparação por igualdade em geral pode ser executada de maneira muito mais eficiente (por exemplo: Merge Join), e
- porque se os atributos comparados de uma das relações forem chave dessa relação, pode-se saber de antemão que a cardinalidade da relação-resultado não será maior do que a dessa relação.
- Isso facilita muito o gerenciamento de memória dos algoritmos que implementam esse operador.
- Como na junção-θ, a relação resultado da equijunção tem todos os atributos de ambas as relações.

Operadores Relacionais Binários

Junção Natural

- Operador de Junção Natural — $\boxed{R_1 * R_2}$, $\boxed{R_1 \bowtie R_2}$ ou $\boxed{R_1 \underset{}{\overset{R_1.a_1, R_2.a_1}{*}} R_2}$
- O operador de Junção Natural é um caso particular da junção-teta, quando θ é"$=$" e os domínios dos atributos comparados também são iguais.
- No operador de Junção Natural $R_1 * R_2$, os atributos da relação-resultado são:
    - os atributos que aparecem só na relação R_1,
    - concatenados aos atributos de R_1 que aparecem na condição de junção c,
    - concatenados aos atributos que estão só em R_2.

Operadores Relacionais Binários

Equijunção **Versus** Junção Natural

- A Junção Natural somente pode ser usada quando os atributos a serem comparados têm o mesmo domínio.
- Por exemplo, dadas as relações definidas como:

```
CREATE TABLE Prof (
    Nome CHAR(20),
    Idade INT
    );
```

```
CREATE TABLE Aluno (
    Nome CHAR(20),
    Idade CHAR(3)
    );
```

- A seguinte consulta gera uma Equijunção:

```
SELECT Prof.Nome, Aluno.Nome
     FROM Prof, Aluno
     WHERE Prof.Idade=Prof.Idade;
```

- Se houverem tuplas assim:

  Prof={<'Pedro', 20>, ...}   Aluno={<'Mario', '20'>, ...}

  então no resultado haverá a tupla

  $\text{Prof} \overset{Idade=Idade}{\bowtie} \text{Aluno}$={<'Pedro', 20, 'Mario', '20'>, ...}

Operadores Relacionais Binários

Equijunção **Versus** Junção Natural

- Se as relações forem definidas como:

```
CREATE TABLE Prof (           CREATE TABLE Aluno (
   Nome CHAR(20),                Nome CHAR(20),
   Idade INT                     Idade INT
   );                            );
```

- A seguinte consulta gera uma Junção Natural

```
SELECT Prof.Nome, Aluno.Nome
    FROM Prof, Aluno
    WHERE Prof.Idade=Prof.Idade;
```

- Se houverem tuplas assim:

  Prof={<'Pedro', 20>, ...}   Aluno={<'Mario', 20>, ...}

  então no resultado haverá a tupla

  $\text{Prof} \overset{Idade}{*} \text{Aluno}$={<'Pedro', 20, 'Mario'>, ...}

Operadores Relacionais Binários

Junções Internas

- As junções teta, equijunção e junção natural sempre emparelham pares de valores existentes em ambas as relações. Por isso são chamadas Junções Internas.
- Existem situações em que é interessante listar também as tuplas de uma relação que não têm valores correspondentes na outra relação.
- Para isso, existe a chamada junção externa. Ela é semelhante à Junção Natural, porém os valores de uma relação que não podem ser emparelhados com os valores na outra relação são repassados para o resultado com valor nulo nos atributos oriundos da outra relação.

Operadores Relacionais Binários

Junções Externas

- O símbolo da Junção externa é: $\boxed{R_1 \overset{R_1.A=R_2.B}{⟗} R_2}$
  Essa é a chamada Junção Externa Completa (*full outer join*).
- Se for necessário que apareçam as tuplas sem correspondência de apenas uma das relações, podem ser usadas:
- Junção Externa à Esquerda (*left outer join*), $\boxed{R_1 \overset{R_1.A=R_2.B}{⟕} R_2}$ ou
- Junção Externa à Direita (*right outer join*), $\boxed{R_1 \overset{R_1.A=R_2.B}{⟖} R_2}$.

Operadores Relacionais Binários

Junções-θ Externas

- As junções internas já estudadas comparam os atributos com o operador de igualdade, portanto podem corresponder tanto a Junções Naturais quanto a Equi-Junções.
- É possível realizar também Junções-θ externas (incluindo as 3 variantes: completa, a esquerda e a direita):
  - Inicialmente, realiza-se a junção-θ normal;
  - Para cada tupla da(s) relações adequadas que não tenha contribuído com uma tupla para o resultado, acrescenta-se essa tupla no resultado, colocando nulo nos atributos que seriam provenientes da outra relação.

Propriedades dos Operadores Relacionais Binários

Propriedades dos Operadores Relacionais Binários — $\times$

Propriedades do operador Produto Cartesiano:

- Do ponto de vista da Álgebra Relacional, o operador de Produto Cartesiano é comutativo, isto é assumindo que $R_1 = \{A\}$ e $R_2 = \{B\}$ então $R_1 \times R_2 = \{A\} \cup \{B\}$ e portanto vale a propriedade comutativa: $R_1 \times R_2 = R_2 \times R_1$
- O operador de Produto Cartesiano é associativo: $(R_1 \times R_2) \times R_3 = R_1 \times (R_2 \times R_3)$
- O operador de Produto Cartesiano não é idempotente: $R_1 \times R_1 \neq R_1$

Propriedades dos Operadores Relacionais Binários

Propriedades do operador de Junção — $\bowtie$

Propriedades do operador de Junção:

- Os três operadores de Junção são comutativos:
  $R_1 \overset{c}{\bowtie} R_2 \Leftrightarrow R_2 \overset{c}{\bowtie} R_1$
- O operador de Junção é associativo:
  Se os atributos envolvidos nas comparações c_1 estão apenas nas relações R_1 e R_2, e os atributos envolvidos nas comparações c_2 estão apenas nas relações R_2 e R_3, então
  $\left(R_1 \overset{c_1}{\bowtie} R_2\right) \overset{c_2}{\bowtie} R_3 \Leftrightarrow R_1 \overset{c_1}{\bowtie} \left(R_2 \overset{c_2}{\bowtie} R_3\right)$

Propriedades dos Operadores Relacionais Binários

Distributividade entre Operadores Unários e Binários — π, σ e ρ

Distributividade entre Operadores Unários ($\boxed{\pi}$, $\boxed{\sigma}$ e $\boxed{\rho}$) e Operadores Binários ($\boxed{\bigcup}$ $\boxed{\bigcap}$ $\boxed{-}$ $\boxed{\times}$ e $\boxed{\bowtie}$:)

- Seja $f(A)$ um operador unário $f \in \{\sigma, \pi \text{ e } \rho\}$
  Seja $\bigcirc$ um operador binário $\bigcirc \in \{\bigcup, \bigcap, -, \times \text{ ou } \bowtie\}$
  Então, as propriedades da distributividade que seguem a forma $f(R_1 \bigcirc R_2) \Leftrightarrow f(R_1) \bigcirc f(R_2)$ podem ser usadas para "antecipar" ou "atrasar" uma seleção ou projeção em relação ao operador binário, **desde que os operadores unárias incorporem as condições/atributos necessários ao operador binário.**
- Veja que para ser possível utilizar $\bigcup$, $\bigcap$ e $-$ é necessário que as relações envolvidas sejam compatíveis em domínio.

Propriedades dos Operadores Relacionais Binários

Distributividade entre Operadores Unários e Binários — π e σ

- As seguintes regras são válidas:

$$\sigma_{(A)}\,(R_1 \bigcup R_2) \Leftrightarrow (\sigma_{(A)} R_1) \bigcup (\sigma_{(A)} R_2)$$
$$\sigma_{(A)}\,(R_1 \bigcap R_2) \Leftrightarrow (\sigma_{(A)} R_1) \bigcap (\sigma_{(A)} R_2)$$
$$\sigma_{(A)}\,(R_1 - R_2) \Leftrightarrow (\sigma_{(A)} R_1) - (\sigma_{(A)} R_2)$$
$$\sigma_{(A)}\,(R_1 \times R_2) \Leftrightarrow (\sigma_{(A)} R_1) \times (\sigma_{(A)} R_2)$$
$$\sigma_{(A)}\,(R_1 \bowtie R_2) \Leftrightarrow (\sigma_{(A)} R_1)\ \bowtie\ (\sigma_{(A)} R_2)$$

$$\pi(A)\,(R_1 \bigcup R_2) \Leftrightarrow (\pi_{(A)} R_1) \bigcup (\pi_{(A)} R_2)$$
|
|
$$\pi(A)\,(R_1 \times R_2) \Leftrightarrow (\pi_{(A)} R_1) \times (\pi_{(A)} R_2)$$
$$\pi(A)\,(R_1 \bowtie R_2) \Leftrightarrow (\pi_{(A)} R_1)\ \bowtie\ (\pi_{(A)} R_2)$$

- Note que a Projeção não distribui sobre a intersecção nem sobre a diferença.
- Note-se que as operações tradicionais sobre escalares ($+$, $-$, $*$, $/$ e potenciação), bem como as operações booleanas, especialmente sobre predicados, continuam válidas e podem ser usadas na transformação de expressões na álgebra relacional.

Propriedades dos Operadores Relacionais Binários

Distributividade entre Operadores Unários e Binários — ρ

- O operador de renomear atributos é distributivo sobre todos os operadores binários

$$\rho_{\{a\backslash b\}}\left(R_1 \bigcup R_2\right) \Leftrightarrow \left(\rho_{\{a\backslash b\}} R_1\right) \bigcup \left(\rho_{\{a\backslash b\}} R_2\right)$$
$$\rho_{\{a\backslash b\}}\left(R_1 \bigcap R_2\right) \Leftrightarrow \left(\rho_{\{a\backslash b\}} R_1\right) \bigcap \left(\rho_{\{a\backslash b\}} R_2\right)$$
$$\rho_{\{a\backslash b\}}\left(R_1 - R_2\right) \Leftrightarrow \left(\rho_{\{a\backslash b\}} R_1\right) - \left(\rho_{\{a\backslash b\}} R_2\right)$$
$$\rho_{\{a\backslash b\}}\left(R_1 \times R_2\right) \Leftrightarrow \left(\rho_{\{a\backslash b\}} R_1\right) \times \left(\rho_{\{a\backslash b\}} R_2\right)$$
$$\rho_{\{a\backslash b\}}\left(R_1 \bowtie R_2\right) \Leftrightarrow \left(\rho_{\{a\backslash b\}} R_1\right) \bowtie \left(\rho_{\{a\backslash b\}} R_2\right)$$

- Note que as propriedades valem desde que não exista conflito entre os nomes, tal como dois atributos com o mesmo nome.

Propriedades dos Operadores Relacionais Binários

Operadores *n*-ádicos

- Operadores binários podem também ser também chamados diádicos, ou 2-ádicos.
- Operadores diádicos com as propriedades de comutatividade e associatividade podem ser generalizados como operadores *n*-ádicos, ou seja, o operador pode ser aplicado em qualquer ordem sobre um conjunto de qualquer cardinalidade de operadores.
- Por exemplo, a união é *n*-ádica, portanto:
  dadas n relações $R_i, i = 1 \ldots n$, então:
  $R_1 \cup R_2 \cup \ldots \cup R_n \Leftrightarrow \bigcup_{i=1}^{n} R_i$.
- Os operadores $\boxed{\sigma}$, $\boxed{\bigcup}$ $\boxed{\bigcap}$ $\boxed{\times}$ e $\boxed{*}$: são *n*-ádicos.

Operadores Relacionais Binários

Divisão

- Operador de Divisão — $R_1 \div R_2$
- Outro Operador Relacional Binário, sem uma ligação intuitiva direta com o produto cartesiano (embora sua definição conceitual o utilize), é a chamada Divisão:
  $T \leftarrow R_1 \div R_2$
- A relação R_2 deve ter como atributos um subconjunto dos atributos da relação R_1, ou seja:
  $R_3(A) \leftarrow R_1(A \cup B) \div R_2(B)$
- A operação de divisão pode ser intuitivamente percebida como uma divisão inteira, em que se buscam os registros $R_3(A)$ cujos valores $R_1(B)$ ocorrem juntamente com todos os valores $R_2(B)$.
- Isto é: para cada valor $R_3(A)$ existe uma sub-relação $R_2(B)$ completa em $R_1(A \cup B)$.

Os Operadores Relacionais Binários como operadores dependentes

- Lembre-se que os operadores {União $\bigcup$, Diferença $-$, Seleção σ, Projeção π, Produto Cartesiano $\times$ e Renomear ρ} são os únicos operadores realmente necessários, o conjunto independente de operadores, já que todos os demais podem ser escritos a partir destes.
- No entanto, os demais operadores, além de serem práticos para expressar consultas, podem ter algoritmos que executam de maneira muito mais eficiente do que pela execução combinada dos operadores independentes, e são muito usados em SGBDs reais.
- Entre os operadores dependentes estão: Intersecção $\bigcap$, União exclusiva $\bigcup|$, Complemento $\neg$, Complemento Ativo $\neg^*$ e todas os Operadores Relacionais Binários, incluindo as várias formas de junções internas (natural $*$, equijoin $\overset{=}{\bowtie}$, e junção-θ $\overset{\theta}{\bowtie}$) e externas (⟗ , ⟖ e ⟕), bem como a Divisão $\div$.

Os Operadores Relacionais Binários como operadores dependentes

- Já foi visto anteriormente que o operador de intersecção pode ser definido em termos dos operadores de União e Diferença.
- Intersecção: $\boxed{R_1 \bigcap R_2 \Leftrightarrow R_1 - (R_1 - R_2)}$
- Da mesma maneira, todos os operadores dependentes podem ser definidos a partir dos operadores independentes:
- União exclusiva $\boxed{R_1 \bigcup | R_2 \Leftrightarrow (R_1 \bigcup R_2) - \big(R_1 - (R_1 - R_2)\big)}$

Os Operadores Relacionais Binários como operadores dependentes

Todas as junções internas podem ser definidas a partir dos operadores independentes:

- Junção Natural — seja $A = \{a_1, a_2, \dots a_i\}$ o conjunto de todos os atributos que aparecem em R_1 e em R_2, seja B o conjunto de todos os atributos que aparecem em R_1 ou em R_2 mas não em ambos e seja $\varphi = (R_1.a_1 = R_2.a_1) \wedge (R_1.a_2 = R_2.a_2) \dots \wedge (R_1.a_i = R_2.a_i)$

  Então: $\boxed{R_1 \overset{A}{*} R_2 \Leftrightarrow \pi_{\{A\}}(\sigma_{\varphi}(R_1 \times R_2))}$

- Equi-Junção — $\boxed{R_1 \overset{R_1.A=R_2.A}{\bowtie} R_2 \Leftrightarrow \sigma_{(R_1.A=R_2.B)}(R_1 \times R_2)}$

- Junção-θ — $\boxed{R_1 \overset{R_1.A\theta R_2.B}{\bowtie} R_2 \Leftrightarrow \sigma_{(R_1.A\theta R_2.B)}(R_1 \times R_2)}$

Os Operadores Relacionais Binários como operadores dependentes

As junções externas também podem ser definidas a partir dos operadores independentes:

- Seja:
  $A = \{r_1, r_2, \ldots r_i\}$ o conjunto de todos os atributos da relação R_1,
  $B = \{s_1, s_2, \ldots s_j\}$ o conjunto de todos os atributos da relação R_2,
  $\perp_1$ uma relação de uma tupla com $|R_1|$ atributos com valor nulo $\perp$ e
  $\perp_2$ uma relação de uma tupla com $|R_2|$ atributos com valor nulo $\perp$.
- Junção externa a Esquerda
  $\boxed{R_1 ⟕ R_2 \Leftrightarrow (R_1 \bowtie R_2) \bigcup ((R_1 - \pi_A(R_1 \bowtie R_2)) \times \perp_2)}$
- Junção externa a Direita
  $\boxed{R_1 ⟖ R_2 \Leftrightarrow (R_1 \bowtie R_2) \bigcup (\perp_1 \times (R_2 - \pi_B(R_1 \bowtie R_2)))}$
- Junção externa Completa
  $\boxed{R_1 ⟗ R_2 \Leftrightarrow (R_1 \bowtie R_2) \bigcup ((R_1 - \pi_A(R_1 \bowtie R_2)) \times \perp_2) \bigcup (\perp_1 \times (R_2 - \pi_B(R_1 \bowtie R_2)))}$

As Operações Relacionais Binárias como operações dependente

Não existe um comando para indicar o operador Divisão em SQL. Assim, ele deve ser expresso pela sua definição:

- Seja:
  A o conjunto dos atributos da relação R_1 que não estão em R_2 e
  B o conjunto de todos os atributos da relação R_2.
  Então R_1 deve ter todos os atributos $A \cup B$ e:
- Divisão $\boxed{R_1 \div R_2 \Leftrightarrow (\pi_A R_1) - \pi_A\Big(\big((\pi_A R_1) \times R_2\big) - R_1\Big)}$
-
```
SELECT DISTINCT A FROM R1
      EXCEPT
SELECT A FROM(
   SELECT A, B FROM (
      (SELECT A FROM R1) AS T1,
      (SELECT B FROM R2) AS T2)
         EXCEPT
   SELECT * FROM R1 AS T3);
```

Outros operadores da Álgebra Relacional

Motivação

- Os SGBDs atuais poderiam se basear apenas nos operadores da álgebra relacional já estudados.
- No entanto, muitas das consultas submetidas pelas aplicações atuais são representadas melhor se forem definidos outros operadores que permitam usar algoritmos mais eficientes do que é possível apenas pela combinação daqueles operadores.
- Assim, é comum o uso de alguns outros operadores, ditos de extensão ao conjunto de operadores básicos

☞ os quais, diga-se de passagem, não aumentam o poder expressivo da álgebra relacional, apenas facilitam a representação de expressões e a busca por alternativas mais eficientes de execução.

Outros operadores da Álgebra Relacional

Motivação

Os operadores que estendem o conjunto básico mais comuns são:

- Semijunção $\ltimes$
- Semidiferença $\triangleright$
- Operadores de Agregação
- Agrupamento γ
- Extensão ψ
- Projeção estendida Π
- Eliminação de Duplicatas τ
- Ordenação ω

Operador de semijunção — $\ltimes$

- O operador de semijunção $\ltimes$ (*semijoin*) é útil especialmente para auxiliar a execução de junções em que são necessários atributos apenas da relação esquerda;
- Isso é muito útil para otimizar consultas, principalmente em ambientes de processamento distribuído de consultas;
- A notação é: $- \boxed{R_1 \overset{R_1.a_1=R_2.a_2}{\ltimes} R_2}$
- e é definido como $R_1 \overset{R_1.a_1=R_2.a_2}{\ltimes} R_2 = \pi_{\{R_1\}}\left(R_1 \overset{R_1.a_1=R_2.a_2}{\bowtie} R_2\right)$
- O resultado do operador de semijunção corresponde a obter todas as tuplas da relação R_1 que têm sua contrapartida na relação R_2.

Operador de semijunção — $\ltimes$

- Por exemplo, dadas as Relações de Alunos e das Disciplinas em que eles se matricularam, onde as relações têm o seguinte esquema:

```
Aluno = {Nome, Idade, Curso}
Matricula = {NomeA, Disciplina, Nota}
```

- considere a consulta:
  Listar todas disciplinas em que os alunos de computação se matricularam.

A resposta corresponde à seguinte expressão:

$$Matricula \overset{NomeA=Nome}{\ltimes} \left(\sigma_{(curso=computacao)} Aluno\right)$$

Operador de semijunção — $\ltimes$

Listar todas disciplinas em que os alunos de computação se matricularam:

$$Matricula \overset{NomeA=Nome}{\ltimes} \left(\sigma_{(curso=computacao)} Aluno\right)$$

```
Aluno = {Nome, Idade, Curso} =
       {<Zeca, 25,    computação>,
        <Zico, 18,    eletronica>,
        <Juca, 21,    odontologia>,
        <Tuca, 18,    computação>}
Matricula = {NomeA, Disciplina, Nota}=
           {<Zeca,  SCC111,     8.0>,
            <Zeca,  SCC112,     9.0>,
            <Zico,  SCC112,     8.5>,
            <Juca,  SCC113,     8.0>,
            <Tuca,  SCC114,     7.0>}
```

Resultado:

```
Result=
{NomeA, Disciplina, Nota}=
{<Zeca,  SCC111,     8.0>,
 <Zeca,  SCC112,     9.0>,
 <Tuca,  SCC114,     7.0>}
```

Note-se que a relação `Aluno` é usada para saber quais são os alunos de computação, e neste caso causa a execução de uma seleção interna, mas nenhum de seus atributos vai para o resultado.

Operador de semijunção — $\ltimes$

Propriedades do operador de semijunção — $\ltimes$

Propriedades do operador de semijunção:

- A semijunção pode ser substituída por operadores de junção:

$$
\begin{aligned}
R_1 \overset{c}{\bowtie} R_2 \quad &\Leftrightarrow \quad (R_1 \overset{c}{\ltimes} R_2) \overset{c}{\bowtie} R_2 \\
&\Leftrightarrow \quad R_1 \overset{c}{\bowtie} (R_2 \overset{c}{\ltimes} R_1) \\
&\Leftrightarrow \quad (R_1 \overset{c}{\ltimes} R_2) \overset{c}{\bowtie} (R_2 \overset{c}{\ltimes} R_1)
\end{aligned}
$$

Operador de Semidiferença — $\triangleright$

- O operador de Semidiferença $\triangleright$ (*semidifference*) é também conhecido como Antijoin ou anti-semijoin (o que seria mais correto). Ele é útil também para auxiliar a execução de junções em que são necessários atributos apenas da relação esquerda;
- A notação é: $\boxed{R_1 \overset{R_1.a_1=R_2.a_2}{\triangleright} R_2}$, mas quando é chamado anti-semijoin usa-se também $\bar{\ltimes}$.
- Ele é definido como $R_1 \overset{R_1.a_1=R_2.a_2}{\triangleright} R_2 = R_1 - \left(R_1 \overset{R_1.a_1=R_2.a_2}{\ltimes} R_2 \right) = R_1 \overset{R_1.a_1 \neq R_2.a_2}{\ltimes} R_2$
- O resultado do operador de semidiferença corresponde a obter todas as tuplas da relação R_1 que não têm uma contrapartida na relação R_2.

Operador de Semidiferença — $\triangleright$

- Por exemplo, dadas as Relações de Alunos e das Disciplinas em que eles se matricularam, onde as relações têm o seguinte esquema:

```
Aluno = {Nome, Idade, Curso}
Matricula = {NomeA, Disciplina, Nota}
```

- considere a consulta:
  Listar todas disciplinas em que nenhum aluno de computação se matriculou.

A resposta corresponde à seguinte expressão:

$$Matricula \overset{NomeA=Nome}{\triangleright} (\sigma_{(curso=computacao)} Aluno)$$

Operador de Semidiferença — ▷

Listar todas disciplinas em que nenhum aluno de computação se matriculou:

$$Matricula \overset{NomeA=Nome}{\triangleright} \left(\sigma_{(curso=computacao)} Aluno\right)$$

```
Aluno = {Nome, Idade, Curso} =
       {<Zeca, 25,    computação>,
        <Zico, 18,    eletronica>,
        <Juca, 21,    odontologia>,
        <Tuca, 18,    computação>}
Matricula = {NomeA, Disciplina, Nota}=
           {<Zeca,  SCC111,     8.0>,
            <Zeca,  SCC112,     9.0>,
            <Zico,  SCC112,     8.5>,
            <Juca,  SCC113,     8.0>,
            <Tuca,  SCC114,     7.0>}
```

Resultado:

```
Result=
{NomeA, Disciplina, Nota}=
{<Zico,  SCC112,    8.5>,
 <Juca,  SCC113,    8.0>}
```

Operador de Semidiferença — $\triangleright$

- Note-se que se R_1 e R_2 forem compatíveis em domínio, então vale a propriedade:
  $R_1 \triangleright R_2 \Leftrightarrow R_1 - R_2$
- Portanto, o operador diferença $(-)$ é um caso especial da semidiferença.
- Esse é um resultado interessante, pois define um operador que estende a diferença com outro mais genérico, que não restringe a estrutura das relações envolvidas.

Operador de Extensão — ψ

- O operador de Extensão ψ permite acrescentar um atributo a uma relação.
- A notação é: $\boxed{\psi_{\{exp \backslash nome_{novo}\}} R}$
- O resultado do operador de extensão sobre a relação R é uma outra relação que tem todos os atributos de R mais um atributo de nome "$nome_{novo}$" e todas as tuplas de R, sendo que o valor do atributo $nome_{novo}$ é dado pela execução da expressão "*exp*" sobre a respectiva tupla.
- A expressão *exp* pode envolver constantes e atributos de R.
- Por exemplo, suponha que a relação de `Alunos={Nome, Idade}` tem a idade do aluno indicada em meses.
  Então $\psi_{\{Idade/12 \backslash Anos\}} Alunos$
  gera a relação $\{Nome, Idade, Anos\}$ onde o valor do atributo `Anos` é `Idade/12`.

Operador de Extensão — ψ

- Note que se quiséssemos acrescentar mais de um atributo, seria necessário usar o operador ψ várias vezes: $\psi_{\{exp_1 \backslash nome_1\}}(\psi_{\{exp_1 \backslash nome_1\}} \ldots (R)) \ldots$
- Como é muito comum que o operador de extensão seja usado junto com um operador de projeção, para simplificar a expressão dessas situações, é comum usar a notação do operador de extensão na lista de atributos da projeção: $\pi_{\{Z\}}(\psi_{\{exp_1 \backslash nome_1\}}(\psi_{\{exp_2 \backslash nome_2\}} R)) \Leftrightarrow \Pi_{\{Z, \psi_{\{exp_1 \backslash nome_1\}}, \psi_{\{exp_2 \backslash nome_2\}}\}} R$
- Veja que, como o operador de projeção não está só projetando, mas também acrescentando atributos, ele é considerado um **operador de projeção estendido**, e usa-se o símbolo Π ao invés de π.

Operador de Extensão — ψ

- Como é comum ocorrer a mesma necessidade também para renomear vários atributos, a mesma notação pode ser usada com o operador de renomear atributos ρ:
  $\pi_{\{Z\}}\left(\rho_{\{atr_1\backslash nome_1\}}(\psi_{\{exp_2\backslash nome_2\}}R)\right) \Leftrightarrow \Pi_{\{Z,\rho_{\{atr_1\backslash nome_1\}},\psi_{\{exp_2\backslash nome_2\}}\}}R$
- Note-se que o operador de extensão ψ não é uma extensão do operador de renomear atributo ρ, pois ψ cria e dá nome a um atributo, enquanto ρ apenas troca o nome de um atributo que já existe.
- O operador de projeção Π faz mais do que projetar os atributos escolhidos, pois é ele que passa a "montar" a relação resultante.

Operadores de Agregação

- Os Operadores de Agregação não são, a princípio, operadores relacionais, mas escalares.
- Um operador de agregação no modelo relacional é aplicado sobre um atributo de uma relação, e retorna um único valor (a princípio um escalar) como função dos valores que esse atributo assume na relação.
- Ele é um operador unário.
- A função é normalmente uma operação estatística sobre os valores.
- A notação de um operador de agregação é: $f_{agreg}(atr)R$
- Os operadores de agregação f_{agreg} normalmente considerados são `COUNT`, `SUM`, `AVG`, `MIN` e `MAX`.

Operadores de Agregação

- Mas lembre-se, a aplicação de um operador de Agregação sobre uma tabela retorna um escalar.
- Por exemplo, dada a relação

```
Aluno = {Nome, Idade, PeríodoCurso} =
       {<Zeca, 25,      2>,
        <Zico, 18,      4>,
        <Juca, 21,      4>,
        <Tuca, 18,      8>}
```

- então a expressão
  $\mathrm{SUM}(Idade)Aluno$ retorna o valor 82.

Operadores de Agregação

- Os operadores de Agregação se tornam operadores relacionais quando se assume que a coleção de todas as aplicações desse operadores sobre uma relação R retorna uma outra relação gR que contem uma única tupla, com um atributo para cada operador de Agregação empregado.
- O operador Π estendido pode fazer essa transformação.
- Por exemplo, dada a relação

```
Aluno = {Nome, Idade, PeríodoCurso} =
       {<Zeca, 25,      2>,
        <Zico, 18,      4>,
        <Juca, 21,      4>,
        <Tuca, 18,      8>}
```

Então $\Pi_{\{\text{SUM}(Idade)\}} Aluno$ retorna a relação $\{<82>\}$.

Operadores de Agregação

- Caso sejam colocadas várias funções de agregação, dada a mesma relação

```
Aluno = {Nome, Idade, PeriodoCurso} =
       {<Zeca, 25,      2>,
        <Zico, 18,      4>,
        <Juca, 21,      4>,
        <Tuca, 18,      8>}
```

- Então

$\Pi_{\{\mathrm{COUNT}(Nome),\mathrm{AVG}(Idade),\mathrm{MIN}(PeriodoCurso),\mathrm{MAX}(PeriodoCurso)\}} Aluno =$

```
 {<    4,      20.25,     2,             8       >}
```

Operadores de Agregação

Considerando que A é um conjunto não nulo de atributos, os Operadores de Agregação calculam:

- `COUNT(*)` - conta o número de tuplas (não necessariamente distintas) da relação. – Esse é o único operador que não requer argumento;
- `COUNT(`A`)` - conta o número de valores (não necessariamente distintos) desse atributo – Não conta tuplas em que o valor do atributo é nulo;
- `COUNT(DISTINCT `A`)` - conta o número de valores distintos desse atributo – Não conta tuplas em que o valor do atributo é nulo. Pode-se escrever `COUNT(ALL `A`)`, mas esse é o *default*;
- `SUM(`A`)` - soma os valores desse atributo;
- `AVG(`A`)` - Calcula a média dos valores desse atributo nas tuplas cujo valor é não nulo;
- `MIN(`A`)` e `MAX(`A`)` - Obtém o menor (maior) valor do atributo que ocorre em qualquer das tuplas.

Operador de Agrupamento — γ

- Com frequência é necessário obter o resultado de funções de agregação não apenas da tabela inteira, mas de grupos de tuplas da relação, de maneira que queremos um valor de agregação para cada grupo.
- Por exemplo, suponha que queremos saber a média de idade dos alunos de cada período: haverá uma média para os alunos do primeiro período, do segundo, etc.
- O operador que faz o agrupamento das tuplas de uma relação é o Operador de Agrupamento γ, cuja notação é: $\boxed{\gamma_{\{Atr_A grup, Atr_A greg\}} R}$ onde:
  - $Atr_a grup$ é uma lista composta de qualquer número de atributos de R – Estes são os atributos usados para agrupar a relação, e são chamados **Atributos de Agrupamento**;
  - $Atr_a greg$ é uma lista composta de qualquer número de Operadores de Agregação, cada um aplicado a um atributo de R, possivelmente com um nome designado – Os resultados são chamados **Atributos de Agregação**;

Operador de Agrupamento — γ

- Por exemplo, dada a relação

```
Aluno = {Nome, Idade, PeríodoCurso} =
       {<Zeca, 25,      2>,
        <Zico, 18,      4>,
        <Juca, 21,      4>,
        <Tuca, 18,      8>}
```

Então

$$\gamma_{\{PeriodoCurso \backslash Periodo, \mathrm{COUNT}(Nome) \backslash Quantos, \mathrm{AVG}(Idade) \backslash IdadeMedia\}} Aluno =$$

```
   = {Período, Quantos, IdadeMedia} =
     {<    2,       1,           25>,
      <    4,       2,           19.5>,
      <    8,       1,           18>}
```

Operador de Agrupamento — γ

- É Importante notar que:
  - Podem haver tantos **Atributos de Agrupamento** quanto necessários: O operador retorna como um grupo todas as tuplas que têm o mesmo valor para a concatenação dos valores de todos os Atributos de Agrupamento;
  - Podem haver tantos **Atributos de Agregação** quanto necessários. Cada um é aplicado sobre apenas um atributo, mas podem ser colocadas várias funções de agregação repetindo um atributo, e vários atributos podem ocorrer em várias funções.
  - O operador de agrupamento pode ser visto como uma generalização do operador de projeção, quando não existem atributos de agregação, isto é: $\gamma_{\{a_1,a_2,\ldots a_n\}} R \Leftrightarrow \Pi_{\{a_1,a_2,\ldots a_n\}} R$.
  - O resultado do operador γ é uma tabela completamente diferente da original:
    Apenas os atributos de agrupamento e de agregação são colocados nessa tabela, portanto os demais atributos da tabela original (além dos atributos de agregação) não existem na tabela resultado.

Operador de Agrupamento — γ

- O resultado do operador γ é uma tabela completamente diferente da original:
- Por exemplo, considere que, dadas as tabelas
  `Aluno = {Nome, Idade, Curso}` e
  `Matricula = {NomeA, Disciplina, Nota}`
- Queremos obter a média de idade nos alunos de cada disciplina cursada pelos alunos de computação, desde que tenha, pelo menos 3 alunos de computação matriculados.
- A resposta corresponde à seguinte expressão:

$$\pi_{\{Disciplina, IdadeMedia\}}\Big(\sigma_{(NMat \geq 3)}\Big($$

dados agrupados

$$\gamma_{\{Disciplina, \mathrm{COUNT}(Nome) \backslash NMat, \mathrm{AVG}(Idade) \backslash IdadeMedia\}}$$

converte

$$(\sigma_{(curso=computacao)}(Aluno \underset{}{\overset{NomeA=nome}{\bowtie}} Matricula))\Big)\Big)$$

dados originais

Trabalhando com Multi-conjuntos e Listas

- A Álgebra Relacional é definida para trabalhar com relações, isto é, subconjuntos do produto cartesiano dos domínios dos atributos envolvidos.
- Portanto, ela é definida para que as relações sejam **conjuntos de tuplas**.
- No entanto, muitas aplicações requerem que a estrutura de dados subjacente não seja "conjunto", mas outras estruturas.
- Os SGBDs, embora ditos "Relacionais", sempre operam com outras estruturas. Inclusive, a terminologia adotada nem é "Relação", mas "**Tabela**"!

Trabalhando com Multi-conjuntos e Listas

- Duas estruturas são usadas com frequência:

☞ **Multi-conjuntos (multisets):** pode haver mais de uma tupla repetida;

☞ **Listas ordenadas:** garante-se que as tuplas (repetidas ou não) estão em uma determinada ordem.

- Para tratar desses dois tipos de dados, mais dois operadores são definidos
  (os quais, obviamente, nem podem ser ditos serem "Operadores da Álgebra Relacional"!):

☞ Operador de Eliminação de Duplicatas τ

☞ Operador de Ordenação ω

Operador de Eliminação de Duplicatas — τ

- Tratar de tuplas duplicadas é fácil: basta não verificar se o resultado de alguns dos operadores repete resultados (por exemplo, após a aplicação de um operador de projeção).
- Isso pode agilizar o processamento em diversas situações, mas a principal motivação para trabalhar com *multisets* é que as aplicações com frequência precisam disso.
- O operador de Eliminação de Duplicatas τ transforma uma "tabela" R, possivelmente com duplicatas, em uma relação.
- A notação é: $\boxed{\tau(R)}$
- Note-se que todos os demais operadores podem ser "estendidos" com facilidade para trabalhar com *multisets*, mas é importante lembrar que nem todas as propriedades mostradas para cada operador continuam válidas se as "relações" admitirem duplicatas.

Operador de Eliminação de Duplicatas — τ

- Estritamente falando, o operador de Eliminação de Duplicatas τ é desnecessário, pois o operador de agrupamento obtém o mesmo resultado.
- Isto é: seja $A = \{a_1, a_2, \ldots a_n\}$ o conjunto de todos os atributos de R. Então $\tau(R) \Leftrightarrow \gamma_{\{A\}}R$.
- Ou seja, cada tupla de uma relação pode ser um grupo de tuplas duplicadas em um *multiset*.
- No entanto, a execução de $\gamma_{\{A\}}R$ é mais custosa do que a execução de um operador $\tau(R)$ específico para a eliminação de duplicatas.
- Assim, como a necessidade de eliminar duplicatas é frequente, o τ é parte do arsenal de operadores normalmente considerado.

Os operadores de conjunto operando com multiconjuntos

- Para trabalhar com *multisets*, é necessário verificar que o comportamento dos operadores pode mudar um pouco.
- Quanto aos operadores de conjunto básicos ($\bigcup$, $\bigcap$ e $-$):
  - $\bigcup$ A união de dois *multisets* não elimina repetições. Assim, a união de dois conjuntos (sem repetição) resulta em dois elementos na união para todos os elementos que estão nos dois conjuntos. Se um dos operandos tiver n repetições de uma mesma tupla e o outro operando tiver m repetições, aparecerão $m + n$ tuplas no resultado.
  - $\bigcap$ Se um dos operandos da intersecção tiver n repetições de uma mesma tupla e o outro operando tiver m repetições, aparecerão $Min(m, n)$ tuplas no resultado.
  - $-$ Se o operando R_1 da diferença tiver n repetições de uma mesma tupla e o operando R_2 tiver m repetições da mesma tupla, aparecerão $Max(0, n - m)$ tuplas no resultado de $R_1 - R_2$.
- Veja que se R_1 e R_2 forem conjuntos, $R_1 - R_2$ e $R_1 \bigcap R_2$ serão conjuntos, mas $R_1 \bigcup R_2$ será *multiset*.

Outros operadores trabalhando com Multiconjuntos

- π A projeção de *multisets* mantém as mesmas propriedades. Veja que projeções podem gerar duplicatas com frequencia.
- σ A seleção aplicada a um conjunto resulta em um conjunto. A seleção aplicada a um multiconjunto pode resultar em um multiconjunto.
- $\times$ O produto cartesiano de multiconjuntos resulta em um multiconjunto: Se o operando R_1 tiver n repetições de uma tupla r_1 e o operando R_2 tiver m repetições de uma tupla r_2, o resultado terá $n * m$ repetições da tupla que concatena r_1 e r_2.

Operador de Ordenação — ω

- Outra estrutura frequentemente necessária é a lista ordenada.
- O operador de Ordenação ω atende a essa necessidade.
- A notação é: $\boxed{\omega_{\{lista\}} R}$
- O resultado do operador de ordenação é uma lista de tuplas ordenadas pelo valor dos atributos indicados em *lista*, os quais devem ser atributos da relação R.
- Se $lista = \{a, b, c\}$, então o resultado e uma lista de todas as tuplas de R ordenadas pelos valores do atributo a. Dentre as tuplas que têm o mesmo valor do atributo a, ordena-se pelo valor do atributo b, e assim por diante. Se houver mais de uma tupla com os mesmos valores de a, b e c, elas são ordenadas arbitrariamente.

Operador de Ordenação — ω

- Ao contrário da estrutura de dados *multiset*, listas ordenadas têm pouco suporte nos SGBDs atuais.
- Ou seja, se uma ou mais listas ordenadas forem operadas pelos demais operadores, não se oferece garantia que o resultado siga qualquer ordem.
- Por isso, usualmente o operador ω é sempre usado como o penúltimo operador de uma expressão de consulta, sendo seguido apenas pelo operador de projeção final, para o qual se garante que preservar a ordem das tuplas.
- O desenvolvimento de SGBDs adaptados para trabalhar com listas ordenadas é uma importante frente de pesquisa atual, sob a denominação de "*ranked queries*".

Outros operadores da Álgebra Relacional

Resumo

Os operadores que estendem o conjunto básico mais comuns são:

- Junções externas – $\boxed{R_1 \overset{R_1.a_1 \theta R_2.a_2}{⟗} R_2}$, $\boxed{R_1 \overset{R_1.a_1 \theta R_2.a_2}{⟖} R_2}$ e $\boxed{R_1 \overset{R_1.a_1 \theta R_2.a_2}{⟕} R_2}$
- Divisão – $\boxed{R_1 \div R_2}$
- Semijunção – $\boxed{R_1 \overset{R_1.a_1 = R_2.a_2}{\ltimes} R_2}$
- Semidiferença – $\boxed{R_1 \overset{R_1.a_1 = R_2.a_2}{\triangleright} R_2}$
- Operadores de Agregação – $\boxed{f_{agreg}(\text{atr-list})R}$
- Agrupamento – $\boxed{\gamma_{\{Lista\}} R}$
- Extensão – $\boxed{\psi_{\{\text{exp}\backslash nome_{novo}\}} R}$
- Projeção estendida – $\boxed{\Pi_{\{Lista\}} R}$
- Eliminação de Duplicatas – $\boxed{\tau(R)}$
- Ordenação – $\boxed{\omega_{\{lista\}} R}$

Outros operadores da Álgebra Relacional

Motivação

- A tradução dos comandos da Linguagem SQL é feita para uma árvore de comandos onde os operadores são aqueles da Álgebra Relacional.
- A seguir, serão dados exemplos de como cada operador da Álgebra Relacional são representados em construções em SQL.

Operadores da Álgebra Relacional Representados em SQL

- Renomear relações e atributos – $\rho_{\{Lista\}} R$
- União – $R_1 \bigcup R_2$
- Intersecção – $R_1 \bigcap R_2$
- Diferença – $R_1 - R_2$
- União exclusiva – $R_1 \bigcup | R_2$
- Complemento – $\neg R$
- Complemento Ativo – $\neg^* R$
- Produto Cartesiano – $R_1 \times R_2$
- Seleção – $\sigma_{(cond)} R$
- Projeção – $\pi_{\{Lista\}} R$
- Junção natural – $R_1 * R_2$
- Equijoin – $R_1 \underset{R_1.a_1 = R_2.a_2}{\bowtie} R_2$
- Junção-θ – $R_1 \underset{R_1.a_1 \theta R_2.a_2}{\bowtie} R_2$
- Junções externas – $R_1 \underset{R_1.a_1 \theta R_2.a_2}{⟗} R_2$, $R_1 \underset{R_1.a_1 \theta R_2.a_2}{⟖} R_2$ e $R_1 \underset{R_1.a_1 \theta R_2.a_2}{⟕} R_2$
- Divisão – $R_1 \div R_2$
- Semijunção – $R_1 \underset{R_1.a_1 = R_2.a_2}{\ltimes} R_2$
- Semidiferença – $R_1 \underset{R_1.a_1 = R_2.a_2}{\rhd} R_2$
- Extensão – $\psi_{\{exp \backslash nome_{novo}\}} R$
- Operadores de Agregação – $f_{agreg}(\text{atr-list}) R$
- Agrupamento – $\gamma_{\{Lista\}} R$
- Projeção estendida – $\Pi_{\{Lista\}} R$
- Eliminação de Duplicatas – $\tau(R)$
- Ordenação – $\omega_{\{lista\}} R$

Operadores da Álgebra Relacional Representados em SQL

- A leitura das Relações é executada por um operador de leitura $\boxed{\text{Le}}$ R_1
- Este não é um operador algébrico, corresponde apenas à leitura de uma tabela, portanto representa um operando Relvar.

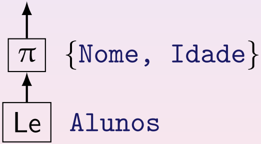
```
SELECT *
   FROM Alunos;
```

$\boxed{\text{Le}}$ `Alunos`

Operador Projeção π

- Operador de Projeção – $\pi_{\{<lista\ de\ atributos>\}} R$

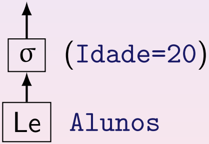
```
SELECT Nome, Idade
   FROM Alunos;
```



Operador Seleção σ

- Operador de Seleção – $\sigma_{(<condicão>)} R$

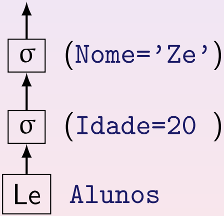
```
SELECT *
   FROM Alunos
   WHERE Idade=20;
```

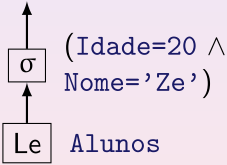


Operador Seleção σ

- Operador de Seleção – $\sigma_{(<condic\tilde{a}o>)}R$

```
SELECT *
   FROM Alunos;
   WHERE Idade=20
   AND Nome='Ze';
```

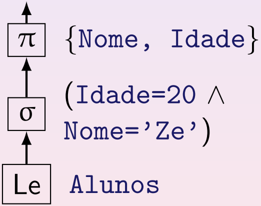




Operador Seleção σ

- Operador de Seleção – $\sigma_{(<condicão>)} R$

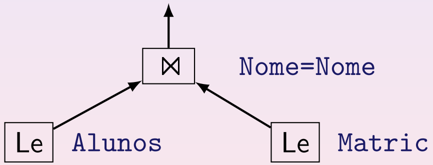
```
SELECT Nome, Idade
   FROM Alunos
   WHERE Idade=20
   AND Nome='Ze';
```



Operador Equijunção $\bowtie$

- Operador de Equijunção — $\boxed{R \overset{AtrR=AtrS}{\bowtie} S}$

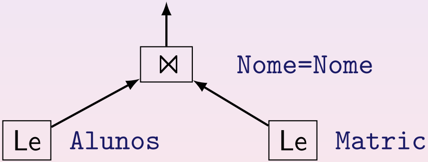
```
SELECT *
   FROM Alunos, Matric
   WHERE Alunos.Nome =
         Matric.Nome;
```



Operador Equijunção $\bowtie$

- Operador de Equijunção — $\boxed{R \overset{AtrR=AtrS}{\bowtie} S}$

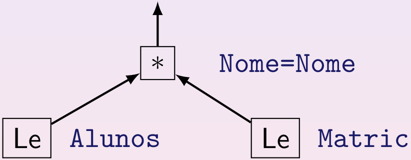
```
SELECT *
    FROM Alunos
       JOIN Matric
         ON Nome=Nome;
```



Operador Junção Natural $*$

- Operador de Junção Natural — $\boxed{R_1 \overset{Atr}{ast} R_2}$

```
SELECT *
   FROM Alunos
       JOIN Matric
       USING Nome;
```

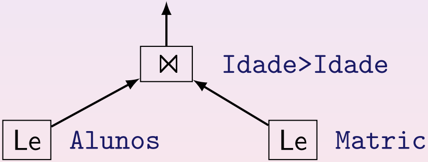


- Veja que indicar `NATURAL JOIN` é opcional nessa construção, uma vez que a subcláusula `USING` já indica essa junção.

Operador Junção-θ $\bowtie$

- Operador de Junção-θ — $\boxed{R \overset{AtrR\theta AtrS}{\bowtie} S}$

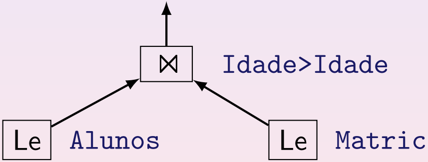
```
SELECT *
    FROM Alunos
       JOIN Matric
         ON Idade>Idade;
```



Operador Junção-θ $\bowtie$

- Operador de Junção-θ — $\boxed{R \underset{}{\overset{AtrR\theta AtrS}{\bowtie}} S}$

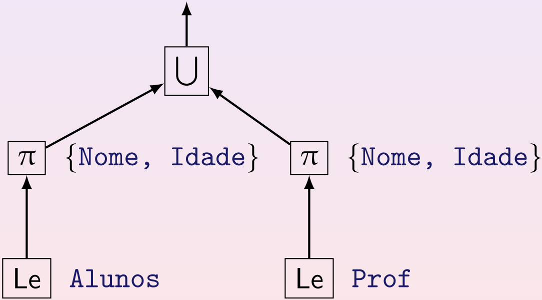
```
SELECT *
   FROM Alunos, Matric
   WHERE Alunos.Idade >
         Matric.Idade;
```



Operador União $\bigcup$

- Operador União: — $\boxed{R_1 \bigcup R_2}$

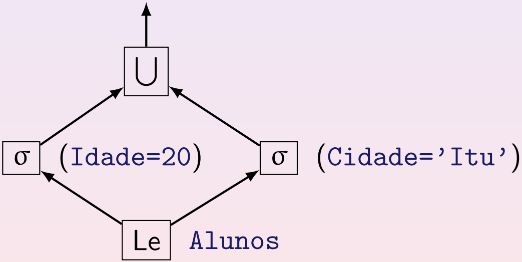
```
SELECT Nome, Idade
    FROM Alunos
  UNION ALL
SELECT Nome, Idade
    FROM Prof;
```



Operador União $\bigcup$

- Operador União: — $\boxed{R_1 \bigcup R_2}$

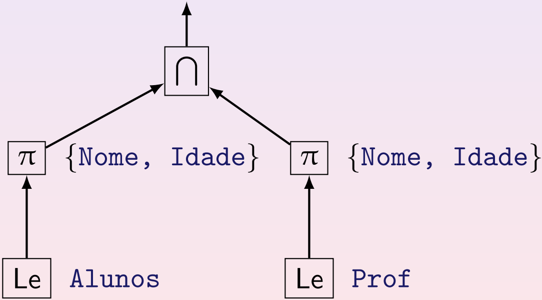
```
SELECT *
   FROM Alunos;
   WHERE Idade=20
   OR Cidade='Itu';
```



Operador Intersecção $\bigcap$

- Operador Intersecção: — $\boxed{R_1 \bigcap R_2}$

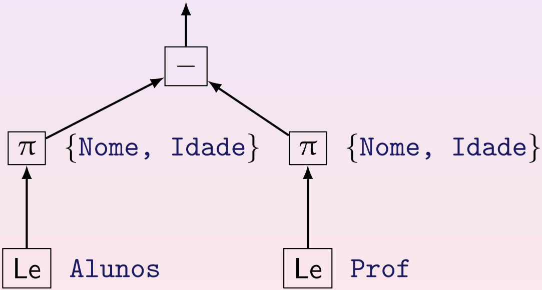
```
SELECT Nome, Idade
    FROM Alunos
  INTERSECT ALL
SELECT Nome, Idade
    FROM Prof;
```



Operador Diferença −

- Operador Diferença: — $\boxed{R_1 - R_2}$

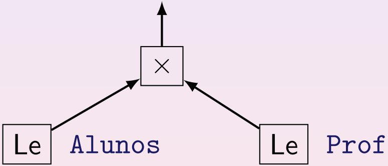
```
SELECT Nome, Idade
    FROM Alunos
  EXCEPT ALL
SELECT Nome, Idade
    FROM Prof;
```



Operador Produto Cartesiano $\times$

- Produto Cartesiano: — $\boxed{R_1 \times R_2}$

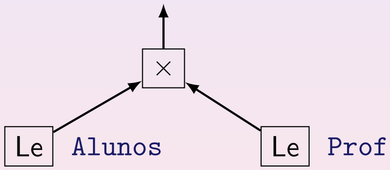
```
SELECT *
    FROM Alunos
       CROSS JOIN Prof;
```



Operador Produto Cartesiano $\times$

- Produto Cartesiano: — $\boxed{R_1 \times R_2}$

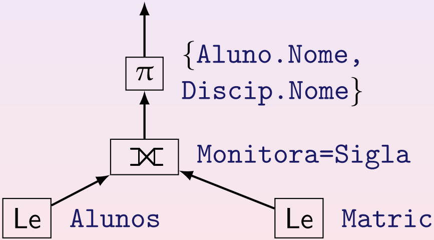
```
SELECT *
    FROM Alunos, Prof;
```



Operador Junção Externa Completa ⟗

- Operador Junção Externa Completa — $\boxed{\overset{AtrR=AtrS}{R_1 ⟗ R_2}}$

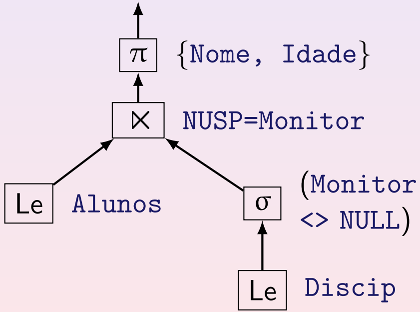
```
SELECT Aluno.Nome,
       Discip.Nome
    FROM Alunos
       OUTER JOIN Discip
       ON Monitora=Sigla;
```



Operador Semijunção $\ltimes$

- Operador Semijunção — $\boxed{R_1 \overset{AtrR=AtrS}{\ltimes} R_2}$

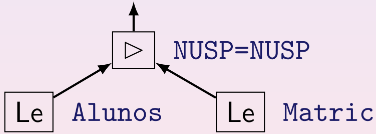
```
SELECT Nome, Idade,
   FROM Alunos
   WHERE NUSP IN (
     SELECT Monitor
     FROM Discip
     WHERE Monitor <> NULL
     );
```



Operador Semidiferença $\triangleright$

- Operador Semidiferença — $\boxed{R_1 \overset{AtrR=AtrS}{\triangleright} R_2}$

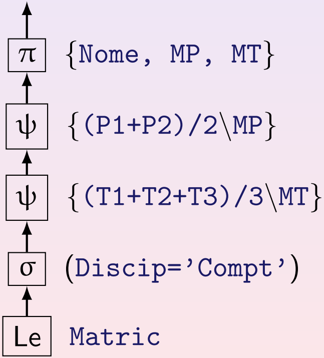
```
SELECT Nome, Idade,
   FROM Alunos
   WHERE NUSP NOT IN (
      SELECT NUSP
      FROM Matric);
```



Operador Extensão ψ

- Operador Extensão – $\boxed{\psi_{\{\text{exp}\backslash nome_{novo}\}} R}$

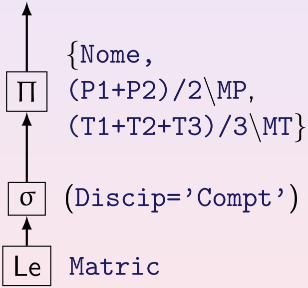
```
SELECT Nome,
     (P1+P2)/2 AS MP,
     (T1+T2+T3)/3 AS MT
    FROM Matric
    WHERE Discip='Compt';
```



Projeção estendida Π

- Projeção estendida – $\boxed{\Pi_{\{Lista\}} R}$

```
SELECT Nome,
     (P1+P2)/2 AS MP,
     (T1+T2+T3)/3 AS MT
   FROM Matric
   WHERE Discip='Compt';
```

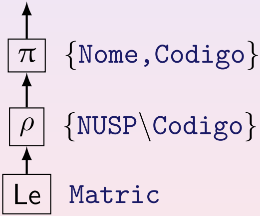


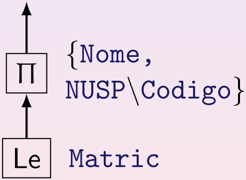
- O mesmo é feito para o operador de renomear atributos.

Renomear Atributos ρ

- Operador Renomear Atributos – $\boxed{\rho_{\{Lista\}} R}$

```
SELECT Nome,
   NUSP AS Codigo,
  FROM Matric
```

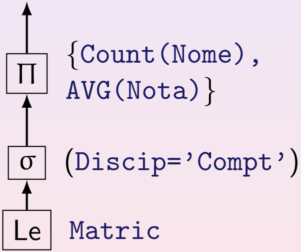




Operadores de Agregação f_{agreg}

- Operadores de Agregação – $\boxed{f_{agreg}(\text{atr-list})R}$

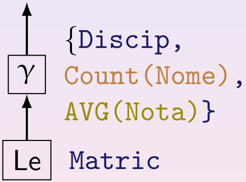
```
SELECT Count(Nome),
     AVG(Nota)
    FROM Matric
    WHERE Discip='Compt';
```



Operador de Agrupamento γ

- Operador de Agrupamento – $\boxed{\gamma_{\{Lista\}} R}$

```
SELECT Discip,
     Count(NUSP)
     AVG(Nota)
    FROM Matric
    GROUP BY Discip;
```

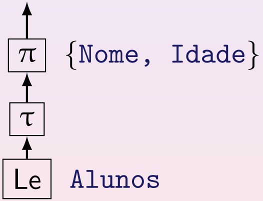


- Note que a lista de atributos do operador de agrupamento inclui dois conjuntos:
  - Os atributos do agrupamento;
  - Os atributos resultantes de funções de agregação.
- Ambos os conjuntos podem ser nulos.

Operador de Eliminação de Duplicatas τ

- Operador de Eliminação de Duplicatas – $\boxed{\tau(R)}$

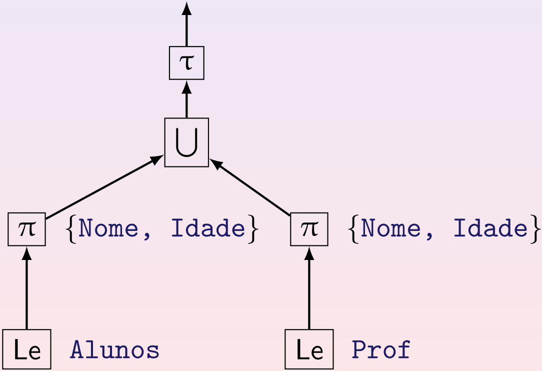
```
SELECT DISTINCT Nome, Idade
   FROM Alunos;
```



Operador de Eliminação de Duplicatas τ

- Operador de Eliminação de Duplicatas – $\boxed{\tau(R)}$

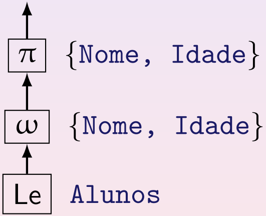
```
SELECT Nome, Idade
    FROM Alunos
  UNION -- sem ALL
SELECT Nome, Idade
    FROM Prof;
```



Operador de Ordenação ω

- Operador de Ordenação – $\boxed{\omega_{\{lista\}} R}$

```
SELECT DISTINCT Nome, Idade
   FROM Alunos
   ORDER BY Nome, Idade;
```



Bases de Dados
–Uma Recordação da Algebra Relacional–

Caetano Traina Jr.

Grupo de Bases de Dados e Imagens
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos
caetano@icmc.usp.br

7 de março de 2013
São Carlos, SP - Brasil

FIM