Banco de Dados
– Álgebra Relacional –

Prof. Dr. Ives Renê V. Pola
ivesr@utfpr.edu.br

Departamento Acadêmico de Informática – DAINF
UTFPR – Pato Branco DAINF
UTFPR
Pato Branco - PR

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.

Roteiro

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 para inclusão ou modificação de tuplas, nem 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 que cada operador corresponde a um **algoritmo** que executa aquele operador sobre a base de dados armazenada num computador ☞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 $*$
  - Equijunção $\overset{=}{\bowtie}$
  - Junção-θ $\overset{\theta}{\bowtie}$
  - Junções externas (⟗ , ⟖ e ⟕)
  - Divisão $\div$

Nomeação de Nomes para Relações e Atributos

- Além dos Operadores Relacionais, a álgebra relacional utiliza ainda o operador "Renomear", simbolizado por $\boxed{\rho}$, para renomear o nome da relação ou os nomes dos atributos.
- O operador ρ quando aplicado numa relação T de grau n, é indicado por qualquer uma das três formas a seguir:

$$\rho_{\mathrm{T}_1(B1,B2,\dots,Bn)}(\mathrm{T}) \quad ou \quad \rho_{\mathrm{T}_1}(\mathrm{T}) \quad ou \quad \rho_{(B1,B2,\dots,Bn)}(\mathrm{T})$$

  onde T_1 é o nome da nova relação e $B1, B2, \dots, Bn$ são os novos nomes dos atributos.
- Uma segunda opção é incluir a renomeação diretamente na operação

  $\rho_{\{NomeAntigo \backslash NomeNovo\}}$

Renomeação/Atribuição de Nome a Relações e a Atributos

Exemplo

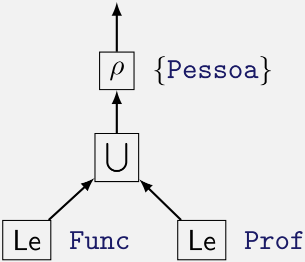
Por exemplo, dado que:

```
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

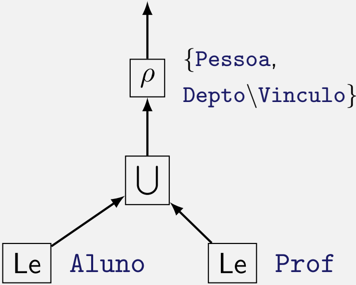
Exemplo

Por exemplo, dado que:

```
Funcionario=(Nome, Idade, Depto)
  Professor=(Nome, Idade, Depto)
```

- Substituição de Nomes de Atributos

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



Renomeação/Atribuição de Nome a Relações e a Atributos

Exemplo

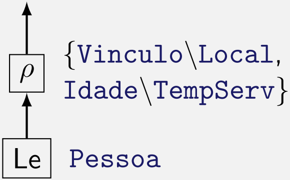
Por exemplo, dado que:

```
    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

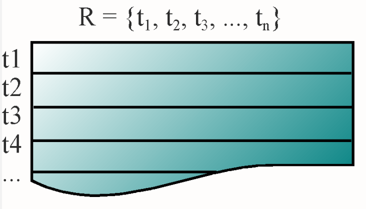
Propriedades do operador Renomear — ρ

Propriedades do operador Renomear:

- O operador de Renomear atributos é idempotente
  $\rho_{\{A_1 \backslash A_2\}}(\rho_{\{A_1 \backslash A_2\}} T) \Leftrightarrow \rho_{\{A_1 \backslash A_2\}} T$
- O operador de Renomear atributos é comutativo
  $\rho_{\{A_1 \backslash A_2\}}(\rho_{\{A_3 \backslash A_4\}} T) \Leftrightarrow \rho_{\{A_3 \backslash A_4\}}(\rho_{\{A_1 \backslash A_2\}} T)$
  (desde que A_1, A_2, A_3 e A_4 não tenham nomes de atributos em comum)

Operadores sobre Conjuntos

- O grupo dos Operadores sobre Conjuntos da Álgebra Relacional corresponde aos 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

Quase todos os Operadores Relacionais sobre Conjuntos são binários.

Pré-requisito

Para que duas relações possam ser operadas por um operador sobre conjunto, é necessário que ambas sejam "Compatíveis em Domínio" (ou "*Union Compatible*") .

Operadores sobre Conjuntos

Relações Compatíveis em Domínio

- Duas relações $T_1(A_1, A_2, ..., A_n)$ e $T_2(B_1, B_2, ..., B_n)$ são ditas Compatíveis em Domínio se ambas têm a mesma dimensionalidade n e se $Dom(A_i) = Dom(B_i), \ \forall i, \ 1 \leq i \leq n$.
- 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.
- Por exemplo:

Dados:

```
Aluno       = (Nome, Idade, Curso)        Dom(nome):  char(30),
Professor   = (Nome, Idade, Depto)        Dom(idade): int,
Funcionário = (Nome, Depto, Idade)        Dom(Curso): char(12),
                                          Dom(Depto): char(12)
```

`Professor` é compatível com `Aluno`, mas não com `Funcionário`.

Operadores sobre Conjuntos

Operadores Unários

- Complemento de uma relação – $\boxed{\neg T}$
  - 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 $\boxed{\neg T = \text{Universo}(T) - T}$.
  - O operador Complemento não é usado em aplicações reais, mas é necessária para a definição da álgebra.
- Complemento ativo de uma relação – $\boxed{\neg^* T}$
  - 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 $\boxed{D\overset{*}{o}m\,(A)\,[T]}$ de um Atributo A numa Relação T 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 T.

Operadores sobre Conjuntos

Operadores binários

- Os operadores relacionais são os usuais da teoria dos conjuntos:
  - União: – $\boxed{T_1 \bigcup T_2}$
    O resultado contém todas as tuplas de T_1 e todas as tuplas de T_2, porém tuplas que estão em ambas as relações aparecem apenas uma vez.
  - Intersecção – $\boxed{T_1 \bigcap T_2}$
    O resultado contém apenas as tuplas que estão em T_1 e também em T_2.
  - Diferença – $\boxed{T_1 - T_2}$
    O resultado contém as tuplas que estão em T_1 mas não estão em T_2.
- Alguns autores também consideram este outro operador:
  - União Exclusiva – $\boxed{T_1 \bigcup | T_2}$ (também chamado Diferença Simétrica)
    O resultado contém todas as tuplas que estão nas relações T_1 ou T_2, mas não as tuplas que estão em ambas T_1 e T_2.

Operadores sobre Conjuntos

Produto Cartesiano

Produto Cartesiano – $\boxed{T_1 \times T_2}$

- O operador “Produto Cartesiano”da álgebra relacional, tal como os demais operadores 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 T_1 e T_2 tem como resultado outra relação em que:
  - Os atributos são a concatenação dos atributos da relação T_1 e da relação T_2,
  - as tuplas são todas as combinações possíveis de valores de T_1 com valores de T_2.

Propriedades dos Operadores sobre Conjuntos

Propriedades dos operadores sobre Conjuntos na Álgebra Relacional

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

- Os operadores de União, Intersecção e União exclusiva são comutativos:
  $T_1 \bigcup T_2 \Leftrightarrow T_2 \bigcup T_1$
  $T_1 \bigcap T_2 \Leftrightarrow T_2 \bigcap T_1$
  $T_1 \bigcup | T_2 \Leftrightarrow T_2 \bigcup | T_1$
  Note-se que $T_1 - T_2 \neq T_2 - T_1$
- Os operadores de União, Intersecção e União exclusiva são associativos:
  $(T_1 \bigcup T_2) \bigcup T_3 \Leftrightarrow T_1 \bigcup (T_2 \bigcup T_3)$
  $(T_1 \bigcap T_2) \bigcap T_3 \Leftrightarrow T_1 \bigcap (T_2 \bigcap T_3)$
  $(T_1 \bigcup | T_2) \bigcup | T_3 \Leftrightarrow T_1 \bigcup | (T_2 \bigcup | T_3)$
  Note-se que $(T_1 - T_2) - T_3 \neq T_1 - (T_2 - T_3)$
- Além disso,
  $T_1 \bigcup T_1 \Leftrightarrow T_1$
  $T_1 \bigcap T_1 \Leftrightarrow T_1$
  Mas $T_1 - T_1 = \emptyset$ e $T_1 \bigcup | T_1 = \emptyset$

Operadores sobre Conjuntos

Propriedades do operador Produto Cartesiano — $\times$

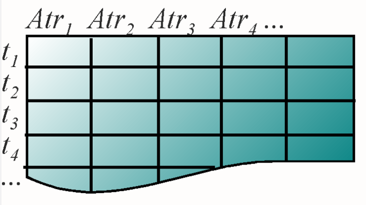
Propriedades do operador Produto Cartesiano:

- Do ponto de vista da Álgebra Relacional, o operador de Produto Cartesiano é comutativo. Isto é, assumindo que $\mathsf{T}_1 = (\mathcal{A})$ e $\mathsf{T}_2 = (\mathcal{B})$ então o esquema de $\mathsf{T}_1 \times \mathsf{T}_2$ é $\{\mathcal{A}\} \cup \{\mathcal{B}\}$ e portanto vale a propriedade comutativa:
  $\mathsf{T}_1 \times \mathsf{T}_2 \Leftrightarrow \mathsf{T}_2 \times \mathsf{T}_1$

  (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:
  $(\mathsf{T}_1 \times \mathsf{T}_2) \times \mathsf{T}_3 \Leftrightarrow \mathsf{T}_1 \times (\mathsf{T}_2 \times \mathsf{T}_3)$
- Mas não é idempotente:
  $\mathsf{T}_1 \times \mathsf{T}_1 \neq \mathsf{T}_1$

Operadores Relacionais Unários

- Os operadores relacionais levam em conta a estrutura interna das relações, reconhecendo 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.
  - $T = (A_1, A_2, \ldots A_n)$
- Existem basicamente 2 operadores relacionais unários:
  - Seleção – $\boxed{\sigma_{(<condic\tilde{a}o>)} T}$
  - Projeção – $\boxed{\pi_{\{<lista\ de\ atributos>\}} T}$



Operadores Relacionais Unários

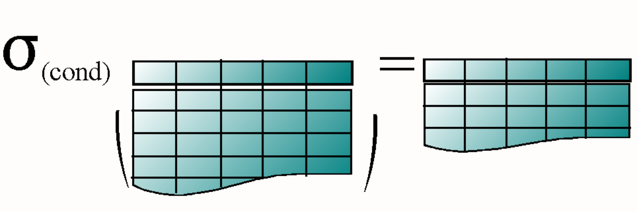
Operador de Seleção

- Operador de Seleção – $\boxed{\sigma_{(c)} \mathrm{T}}$
  - Aplicada sobre uma relação T, o operador de 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_1 da relação T com:
    - Uma constante: $c := \mathrm{T.A}_1 \, \theta \, cte$;
    - Ou com outro atributo da própria relação, sempre comparando os valores de dois atributos da mesma tupla: $c := \mathrm{T.A}_1 \, \theta \, \mathrm{T.A}_2$.
  - O operador de comparação θ é qualquer operador válido no domínio do atributo $\mathrm{T.A}_i$. Tipicamente os operadores de igualdade ($=$ e $\neq$) e relacionais ($>$, $\geq$, $<$, $\leq$) são válidos para qualquer atributo textual, numérico ou datas, e outros podem ser válidos em domínios específicos, tal como continência para textos.

Operadores Relacionais Unários

Operador de Seleção

- Intuitivamente, o operador de Seleção pode ser visto como sendo a escolha de algumas "linhas" da tabela que é a relação.



Operadores Relacionais Unários

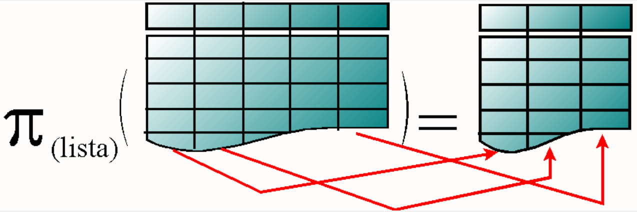
Operador de Projeção

- Operador de Projeção – $\boxed{\pi_{\{<\textit{lista de atributos}>\}}\mathsf{T}}$
  - O operador de projeção aplicado sobre uma relação T 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.

Operadores Relacionais Unários

Operador de Projeção

- Intuitivamente, o operador de Projeção pode ser entendido como a escolha de algumas "Colunas" da tabela.



Operadores Relacionais Unários

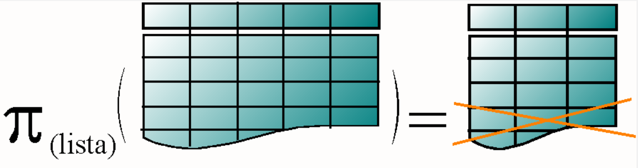
Operador de Projeção

- Considerações sobre o operador de projeção:
  - O resultado de executar um operador de projeção é uma relação, portanto não deve existir tuplas repetidas no resultado;
  - Se a <*lista de atributos*> contiver uma chave da relação, então pode-se ter certeza de que o resultado não terá tuplas repetidas;
  - Se a <*lista de 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 com pelo operador de Eliminação de Tuplas Repetidas, embora obviamente esse não seja um operador "algébrico".

Operadores Relacionais Unários

Operador de Projeção

- Note-se que numa operação de Projeção, além de se descartar algumas "colunas" da tabela, algumas tuplas repetidas podem ser eliminadas, diminuindo também algumas "linhas" da tabela.



Propriedades dos Operadores Relacionais Unários

Propriedades do operador de seleção — σ

Propriedades do operador de seleção:

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

Propriedades dos Operadores Relacionais Unários

Propriedades do operador de seleção — σ

Outras propriedades:

- $\sigma_{(<condic\tilde{a}o_1>\wedge<condic\tilde{a}o_2>)}T \Leftrightarrow \sigma_{(<condic\tilde{a}o_1>)}T \cap \sigma_{(<condic\tilde{a}o_2>)}T$
- $\sigma_{(<condic\tilde{a}o_1>\vee<condic\tilde{a}o_2>)}T \Leftrightarrow \sigma_{(<condic\tilde{a}o_1>)}T \cup \sigma_{(<condic\tilde{a}o_2>)}T$
- Idempotência:
  $\sigma_{(<condic\tilde{a}o_1>)}\left(\sigma_{(<condic\tilde{a}o_1>)}T\right) \Leftrightarrow \sigma_{(<condic\tilde{a}o_1>)}T$

Propriedades dos 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 A_2 contém o subconjunto A_1, então vale a igualdade: $\pi_{\{A_1\}}\left(\pi_{\{A_2\}}T\right) \Leftrightarrow \pi_{\{A_1\}}T$

Propriedades dos 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_1 e A_2 tal que $A_1 \subseteq A_2$ então: $\pi_{\{A_2\}}\left(\sigma_{(A_1)} T\right) \Leftrightarrow \sigma_{(A_1)}\left(\pi_{\{A_2\}} T\right)$

Operadores Relacionais Binários

- Pela Teoria da Álgebra Relacional, se estiverem definidos apenas os seguintes operadores, todos os demais podem ser definidos a partir deles:
  - União $\bigcup$
  - Diferença $-$
  - Produto Cartesiano $\times$
  - Seleção σ
  - Projeção π
  - Renomear relações e atributos ρ
- Esse é o conjunto dos "operadores independentes"
- Por exemplo, o operador de Intersecção é dito ser um operador dependente, porque pode ser definido usando apenas o operador de diferença:
  $T_1 \bigcap T_2 \Leftrightarrow T_1 - (T_1 - T_2)$

Operadores Relacionais Binários

- Alguns operadores 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 — $\boxed{T_1 * T_2}$
  - a Equijunção — $\boxed{T_1 \overset{A_1 = A_2}{\bowtie} T_2}$
  - e a Junção-θ — $\boxed{T_1 \overset{A_1 \theta A_2}{\bowtie} T_2}$

Operadores Relacionais Binários

Junção-θ

- Operador de Junção-θ (ou θ-*join*) — $\boxed{T_1 \overset{c}{\bowtie} T_2}$
- Este é o operador de junção mais genérico e flexível, mas também o mais lento.
- A condição de Junção c segue a forma:
  <condição> $\wedge$ <condição> $\wedge$ ... $\wedge$ <condição>, onde:
  - A <condição> tem a forma $A_i \ \theta \ B_j$
  - A_i pertence à T_1 e B_j pertence à T_2, tendo o mesmo domínio, e
  - θ é um operador de comparação válido nesse domínio.
- Ambos os atributos envolvidos na expressão de comparação c aparecem na relação resultado.
- Tuplas cujos atributos de comparação não tem valores correspondentes nas duas relações não aparecem no resultado.

Operadores Relacionais Binários

Equi-Junção

- Operador de Equijunção — $\boxed{T_1 \underset{T_1.A_1=T_2.A_1}{\bowtie} T_2}$ ou $\boxed{T_1 \underset{T_1.A_1,T_2.A_1}{\bowtie} T_2}$
- Ele é um caso particular da junção-teta, quando θ é "$=$".
    - pode-se simplificar o algoritmo de comparação.
- Eles são tratados de maneira distinta porque a comparação por igualdade permite ter algoritmos muito mais eficientes (por exemplo: *Merge Join*)
    - Se os atributos comparados são chave de uma relação, sabe-se de antemão que a cardinalidade do resultado não será maior do que a dessa relação.
    - Isso facilita o gerenciamento de memória desses algoritmos.
- Ambos os atributos envolvidos na expressão de comparação aparecem no resultado, o que resulta em pares de atributos com valores iguais.
- Tuplas cujos atributos de comparação não tem valores correspondentes nas duas relações não aparecem no resultado.

Operadores Relacionais Binários

Junção Natural

- Operador de Junção Natural — $\boxed{T_1 * T_2}$, $\boxed{\overset{T_1.A_i,T_2.B_j}{T_1 * T_2}}$ ou $\boxed{T_1 \overset{A_1}{*} T_2}$
- Na Equi-Junção, cada par de atributos comparados possuem valores idênticos no resultado.
- O operador de Junção Natural é um caso particular da Equi-Join, eliminando o atributo desnecessário.
  - Escolhendo os atributos chaves da relação T_1,
- Tuplas cujos atributos envolvidos na comparação não tem valores correspondentes nas duas relações não aparecem no resultado.

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 e seus tipos:
  `Prof (Nome CHAR(20), Idade INT )`
  `Aluno (Nome CHAR(20), Idade CHAR(3) )`
- A seguinte consulta gera uma Equi-Junção:
  $$\texttt{Prof} \overset{Idade=Idade}{\bowtie} \texttt{Aluno}$$
- Se houverem tuplas assim:
  `Prof={<'Pedro', 20>, ...}   Aluno={<'Mario', '20'>, ...}`
  então no resultado haverá a tupla
  `{<'Pedro', 20, 'Mario', '20'>, ...}`

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 e seus tipos:
  `Prof (Nome CHAR(20), Idade INT )`
  `Aluno (Nome CHAR(20), Idade INT )`
- A seguinte consulta gera uma Junção Natural:
  $\texttt{Prof} \overset{Idade}{*} \texttt{Aluno}$
- Se houverem tuplas assim:
  `Prof={<'Pedro', 20>, ...}   Aluno={<'Mario', '20'>, ...}`
  então no resultado haverá a tupla
  `{<'Pedro', 20, 'Mario'>, ...}`

Operadores Relacionais Binários

Junções Internas

- As três formas de junção apresentadas somente resultam tuplas onde os pares de valores comparados são existentes e iguais em ambas as relações. Por isso são chamadas Junções Internas (*inner joins*).

  - Junção-θ — $\boxed{T_1 \overset{A_1\,\theta\,A_2}{\bowtie} T_2}$
  - Equi-Junção — $\boxed{T_1 \overset{A_1=A_2}{\bowtie} T_2}$ ou $\boxed{T_1 \overset{A_1,A_2}{\bowtie} T_2}$
  - Junção Natural — $\boxed{T_1 \overset{A}{*} T_2}$ ou $\boxed{T_1 \overset{A_1,A_2}{*} T_2}$

Operadores Relacionais Binários

Junções Internas

- Existem situações em que é interessante listar também no resultado da Junção as tuplas de uma relação que não têm valores correspondentes na outra relação.
- Para isso, existe a operação chamada Junção Externa.
  - Ela é semelhante à Junção Natural, mas os valores de uma relação não relacionados com os valores da 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: $\boxed{\overset{T_1.A_1=T_2.A_2}{T_1 ⟗ T_2}}$

  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*): $\boxed{\overset{T_1.A_1=T_2.A_2}{T_1 ⟕ T_2}}$
- Junção Externa à Direita (*right outer join*): $\boxed{\overset{T_1.A_1=T_2.A_2}{T_1 ⟖ T_2}}$

Operadores Relacionais Binários

Junções-θ Externas

- É possível realizar também Junções-θ externas (incluindo as 3 variantes: completa, a esquerda e a direita):
  - Inicialmente, realiza-se a junção-θ de maneira normal;
  - Durante o processo, para cada tupla da(s) relações que não tenha contribuído com uma tupla no resultado, acrescenta-se essa tupla no resultado, colocando nulo nos atributos que seriam provenientes da outra relação.

Operadores Relacionais Binários

Divisão

- Outro Operador Relacional Binário, é o operador de Divisão:
  $T_3 \leftarrow T_1 \div T_2$
- A relação T_2 deve ter como atributos um subconjunto dos atributos da relação T_1, ou seja:
  $T_3(A_1) \leftarrow T_1(A_1 \cup A_2) \div T_2(A_2)$
- O operador divisão pode ser intuitivamente entendido como uma divisão inteira, em que se buscam os registros $T_3(A_1)$ cujos valores $T_1(A_2)$ ocorrem juntamente com **todos** os valores $T_2(A_2)$.
- Isto é: para cada valor $T_3(A_1)$ existe uma sub-relação $T_2(A_2)$ completa em $T_1(A_1 \cup A_2)$.

Propriedades dos Operadores Relacionais Binários

Propriedades do operador de Junção — $\bowtie$

Propriedades do operador de Junção:

- São comutativos:

  $$T_1 \overset{c}{\bowtie} T_2 \Leftrightarrow T_2 \overset{c}{\bowtie} T_1$$

- São associativos:

  $$\left(T_1 \overset{c_1}{\bowtie} T_2\right) \overset{c_2}{\bowtie} T_3 \Leftrightarrow T_1 \overset{c_1}{\bowtie} \left(T_2 \overset{c_2}{\bowtie} T_3\right)$$

Propriedades dos Operadores Relacionais Binários

Distributividade entre Operadores Unários e Binários — π e σ

- As seguintes regras são válidas:

$$\sigma_{(A)}(T_1 \bigcup T_2) \Leftrightarrow (\sigma_{(A)}T_1) \bigcup (\sigma_{(A)}T_2)$$
$$\sigma_{(A)}(T_1 \bigcap T_2) \Leftrightarrow (\sigma_{(A)}T_1) \bigcap (\sigma_{(A)}T_2)$$
$$\sigma_{(A)}(T_1 - T_2) \Leftrightarrow (\sigma_{(A)}T_1) - (\sigma_{(A)}T_2)$$
$$\sigma_{(A)}(T_1 \times T_2) \Leftrightarrow (\sigma_{(A)}T_1) \times (\sigma_{(A)}T_2)$$
$$\sigma_{(A)}(T_1 \bowtie T_2) \Leftrightarrow (\sigma_{(A)}T_1) \bowtie (\sigma_{(A)}T_2)$$

$$\pi_{\{A\}}(T_1 \bigcup T_2) \Leftrightarrow (\pi_{\{A\}}T_1) \bigcup (\pi_{\{A\}}T_2)$$
$$|$$
$$|$$
$$\pi_{\{A\}}(T_1 \times T_2) \Leftrightarrow (\pi_{\{A\}}T_1) \times (\pi_{\{A\}}T_2)$$
$$\pi_{\{A\}}(T_1 \bowtie T_2) \Leftrightarrow (\pi_{\{A\}}T_1) \bowtie (\pi_{\{A\}}T_2)$$

- Note que a Projeção não distribui sobre a intersecção nem sobre a diferença.
- 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

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 $\mathrm{T}_i, i = 1 \ldots n$, então:
  $\mathrm{T}_1 \cup \mathrm{T}_2 \cup \ldots \cup \mathrm{T}_n \Leftrightarrow \bigcup_{i=1}^{n} \mathrm{T}_i$.
- Os operadores $\boxed{\sigma}$, $\boxed{\bigcup}$ $\boxed{\bigcap}$ $\boxed{\times}$ e $\boxed{*}$: são *n*-ádicos.

Funções de Agregação

- As Funções de Agregação não são operadores relacionais, mas funções que retornam valores escalares.
- Uma função de agregação é aplicada a um atributo de uma relação, e retorna um único valor (a princípio escalar) calculada sobre os valores que esse atributo assume na relação.
- A função é normalmente uma operação estatística sobre os valores.
- A notação de uma função de agregação é: $f_{agreg}(\text{T.A})$
- As funções de agregação f_{agreg} normalmente disponíveis são `COUNT`, `SUM`, `AVG`, `MIN` e `MAX`.

Funções de Agregação

- Mas lembre-se, a aplicação de uma Função de Agregação sobre todas as tuplas de uma tabela retorna um valor.
- Por exemplo, dada a relação

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

- então a expressão

  SUM(*Aluno.Idade*)   retorna o valor 82.

Funções de Agregação

- As Funções de Agregação participam de operadores relacionais de maneira que a coleção de todas as aplicações de funções sobre a relação T cria uma outra relação gT que contem uma única tupla, com um atributo para cada função de Agregação empregada.
- O operador Π estendido participa dessa transformação, junto com um 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 $\gamma\Pi_{\{\mathrm{SUM}(Idade)\}}Aluno$ retorna a relação $\{<82>\}$.

Funções de Agregação

- Caso sejam colocadas várias funções de agregação, a relação resultante terá um atributo para cada função, em um resultado composto por uma tupla.

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

- Então

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

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

Funções de Agregação

Considerando que A é um conjunto não nulo de atributos, as funções de Agregação calculam:

- `COUNT(*)` - conta o número de tuplas (não necessariamente distintas) da relação. – Esse é a única função 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 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: $\boxed{\gamma_{\{\{Atr_{Agrup}\},\{Fn_{Agreg}\}\}}\mathsf{T}}$ onde:
  - $\{Atr_{Agrup}\}$ é uma lista composta de qualquer número de atributos de T – Estes são os atributos usados para agrupar a relação, e são chamados **Atributos de Agrupamento**;
  - $\{Fn_{Agreg}\}$ é uma lista composta de qualquer número de Funções de Agregação, cada um aplicado a um atributo de T, possivelmente com um nome designado – e chamados **Funções 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.

Operador de Agrupamento — γ

- É Importante notar que:
  - 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.
  - O operador de agrupamento pode ser visto como uma generalização do operador de projeção, **SE** não existirem atributos de agregação, isto é: $\gamma_{\{\{A_1,A_2,\ldots A_n\},\emptyset\}} T \Leftrightarrow \Pi_{\{A_1,A_2,\ldots A_n\}} T$.

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)`
  `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:

$$\pi_{\{Disciplina, IdadeMedia\}}\left(\sigma_{(NMat \geq 3)}\left(\right.\right.$$

☜ dados agrupados

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

☜ converte

$$\left.\left.\left(\sigma_{(curso=\text{“computacao”})}\left(Aluno \overset{Nome=nomeA}{\bowtie} Matricula\right)\right)\right)\right)$$

☜ 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

Trabalhando com Multi-conjuntos e Listas

Multi-conjuntos

- Com frequência, Multi-conjuntos são tratados com *bags*.

☞ Porque admitir multi-conjuntos nos SGBDRs?

  - Podem ser mais eficientes: União e Projeção requerem a Eliminação de Duplicatas – que pode ser desnecessária pela semântica da aplicação.
  - Multi-conjuntos permitem que outras operações sejam possíveis.

  ☞ Por exemplo, obter a média dos alunos de uma dada disciplina:
  $\text{AVG}\left(\pi_{\{Nota\}}\left(\sigma_{(Disciplina=\text{``BD''})} Matricula\right)\right)$

  ☞ Se o operador π retornasse um conjunto, essa construção resultaria em uma média errada sempre que uma turma tiver pelo menos dois alunos com a mesma nota.

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" T, possivelmente com duplicatas, em uma relação.
- A notação é: $\boxed{\tau(T)}$
- 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 $\mathcal{A} = \{A_1, A_2, \ldots A_n\}$ o conjunto de todos os atributos de T. Então $\tau(T) \Leftrightarrow \gamma_{\{\mathcal{A}\}}T$.
- Ou seja, cada tupla de uma relação pode ser um grupo de tuplas duplicadas em um *multiset*.
- No entanto, a execução de $\gamma_{\{\mathcal{A}\}}T$ é mais custosa do que a execução de um operador $\tau(T)$ 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 $\max(m, n)$ 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 T_1 da diferença tiver n repetições de uma mesma tupla e o operando T_2 tiver m repetições da mesma tupla, aparecerão $\max(0, n-m)$ tuplas no resultado de $T_1 - T_2$.

Outros operadores trabalhando com Multiconjuntos

- π A projeção de *multisets* mantém as mesmas propriedades. Veja que projeções podem gerar duplicatas com frequência.
- 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 T_1 tiver n repetições de uma tupla t_i e o operando T_2 tiver m repetições de uma tupla t_j, o resultado terá $n * m$ repetições da tupla que concatena t_i e t_j.

Operador de Ordenação — ω

- Outra estrutura frequentemente necessária é a lista ordenada.
- O operador de Ordenação ω atende a essa necessidade.
- A notação é: $\boxed{\omega_{\{lista\}}\mathrm{T}}$
- 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 T.
- Se $lista = (\mathrm{A}_a, \mathrm{A}_b, \mathrm{A}_c)$, então o resultado é uma lista de todas as tuplas de T ordenadas pelos valores do atributo A_a. Dentre as tuplas que têm o mesmo valor do atributo A_a, ordena-se pelo valor do atributo A_b, e assim por diante. Se houver mais de uma tupla com os mesmos valores de $\mathrm{A}_a, \mathrm{A}_b$ e A_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 – $\boxed{T_1 \overset{T_1.A_1 \theta T_2.A_2}{⟗} T_2}$, $\boxed{T_1 \overset{T_1.A_1 \theta T_2.A_2}{⟖} T_2}$ e $\boxed{T_1 \overset{T_1.A_1 \theta T_2.A_2}{⟕} T_2}$
- Divisão – $\boxed{T_1 \div T_2}$
- Operadores de Agregação – $\boxed{f_{agreg}(\text{atr-list})T}$
- Agrupamento – $\boxed{\gamma_{\{\{Lista\},\{f_{agreg}\}\}} T}$
- Projeção estendida – $\boxed{\Pi_{\{Lista\}} T}$
- Eliminação de Duplicatas – $\boxed{\tau(T)}$
- Ordenação – $\boxed{\omega_{\{lista\}} T}$

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 – $\boxed{\rho_{\{Lista\}}\mathrm{T}}$
- União – $\boxed{\mathrm{T}_1 \bigcup \mathrm{T}_2}$
- Intersecção – $\boxed{\mathrm{T}_1 \bigcap \mathrm{T}_2}$
- Diferença – $\boxed{\mathrm{T}_1 - \mathrm{T}_2}$
- União exclusiva – $\boxed{\mathrm{T}_1 \bigcup| \mathrm{T}_2}$
- Complemento – $\boxed{\neg \mathrm{T}}$
- Complemento Ativo – $\boxed{\neg^* \mathrm{T}}$
- Produto Cartesiano – $\boxed{\mathrm{T}_1 \times \mathrm{T}_2}$
- Seleção – $\boxed{\sigma_{(cond)}\mathrm{T}}$
- Projeção – $\boxed{\pi_{\{Lista\}}\mathrm{T}}$
- Junção natural – $\boxed{\mathrm{T}_1 * \mathrm{T}_2}$
- Equijunção – $\boxed{\mathrm{T}_1 \underset{}{\overset{\mathrm{T}_1.\mathrm{A}_1=\mathrm{T}_2.\mathrm{A}_2}{\bowtie}} \mathrm{T}_2}$
- Junção-θ – $\boxed{\mathrm{T}_1 \overset{\mathrm{T}_1.\mathrm{A}_1\theta\mathrm{T}_2.\mathrm{A}_2}{\bowtie} \mathrm{T}_2}$
- Junções externas – $\boxed{\mathrm{T}_1 \overset{\mathrm{T}_1.\mathrm{A}_1\theta\mathrm{T}_2.\mathrm{A}_2}{⟗} \mathrm{T}_2}$, $\boxed{\mathrm{T}_1 \overset{\mathrm{T}_1.\mathrm{A}_1\theta\mathrm{T}_2.\mathrm{A}_2}{⟕} \mathrm{T}_2}$ e $\boxed{\mathrm{T}_1 \overset{\mathrm{T}_1.\mathrm{A}_1\theta\mathrm{T}_2.\mathrm{A}_2}{⟖} \mathrm{T}_2}$
- Divisão – $\boxed{\mathrm{T}_1 \div \mathrm{T}_2}$
- Operadores de Agregação – $\boxed{f_{agreg}(\text{atr-list})\mathrm{T}}$
- Agrupamento – $\boxed{\gamma_{\{\{Lista\},\{f_{agreg}\}\}}\mathrm{T}}$
- Projeção estendida – $\boxed{\Pi_{\{Lista\}}\mathrm{T}}$
- Eliminação de Duplicatas – $\boxed{\tau(\mathrm{T})}$
- Ordenação – $\boxed{\omega_{\{lista\}}\mathrm{T}}$

Operadores da Álgebra Relacional Representados em SQL

- A leitura das Relações é executada por um operador de leitura $\boxed{\text{Le}}\ T_1$
- Este não é um operador algébrico, corresponde apenas à leitura de uma tabela, portanto representa um operando Relvar.

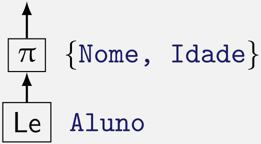
```
SELECT *
   FROM Aluno;
```

$\boxed{\text{Le}}$ Aluno

Operador Projeção π

- Operador de Projeção: $\pi_{\{<\textit{lista de atributos}>\}} T$

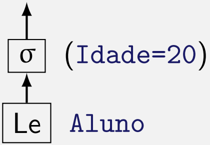
```
SELECT Nome, Idade
   FROM Aluno;
```



Operador Seleção σ

- Operador de Seleção: $\sigma_{(<\text{condição}>)}\mathsf{T}$

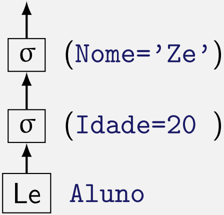
```
SELECT *
    FROM Aluno
    WHERE Idade=20;
```

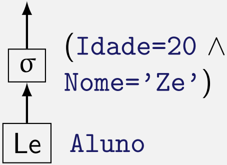


Operador Seleção σ

- Operador de Seleção: $\sigma_{(<\text{condição}>)}T$

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





Operador Seleção σ

- Operador de Seleção: $\sigma_{(<condicão>)}T$

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

$\boxed{\pi}$ {Nome, Idade}

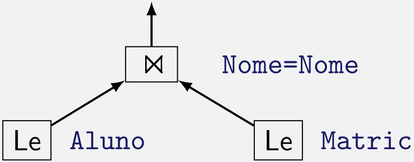
$\boxed{\sigma}$ (Idade=20 $\wedge$ Nome='Ze')

$\boxed{\text{Le}}$ Aluno

Operador Equijunção $\bowtie$

- Operador de Equijunção: $\boxed{T_1 \bowtie_{AtrT_1=AtrT_2} T_2}$

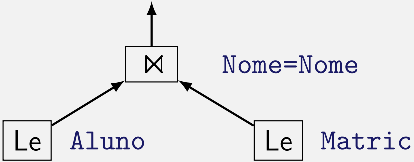
```
SELECT *
   FROM Aluno, Matric
   WHERE Aluno.Nome =
         Matric.Nome;
```



Operador Equijunção $\bowtie$

- Operador de Equijunção: $\boxed{\overset{AtrT_1=AtrT_2}{T_1 \bowtie T_2}}$

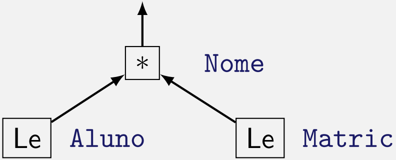
```
SELECT *
    FROM Aluno
       JOIN Matric
         ON Nome=Nome;
```



Operador Junção Natural $*$

- Operador de Junção Natural: $\boxed{T_1 \overset{Atr}{*} T_2}$

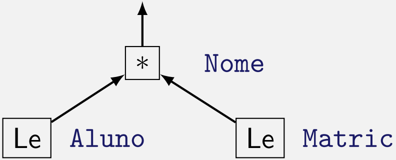
```
SELECT *
   FROM Aluno
       NATURAL JOIN
       Matric;
```



Operador Junção Natural $*$

- Operador de Junção Natural: $\boxed{T_1 \overset{Atr}{*} T_2}$

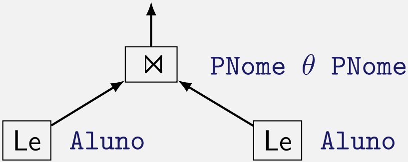
```
SELECT *
   FROM Aluno
       JOIN Matric
       USING Nome;
```



Operador Junção-θ $\bowtie$

- Operador de Junção-θ: $\boxed{T_1 \underset{}{\overset{AtrT_1 \theta AtrT_2}{\bowtie}} T_2}$

```
SELECT *
FROM Aluno A1
   JOIN Aluno A2 ON
   levenshtein(A1.PNome,
   A2.PNome) < 2;
```

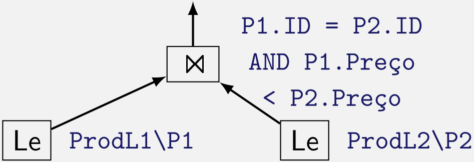


Combinação de Operadores Junção-θ e Equi-Junção $\bowtie$

- Operador de Junção-θ — $\boxed{T_1 \overset{AtrT_1 \theta AtrT_2}{\bowtie} T_2}$
- "Quais produtos a loja 1 vende mais barato que a loja 2?"

```
ProdL1(ID, Nome, Preço);
ProdL2(ID, Nome, Preço);
```

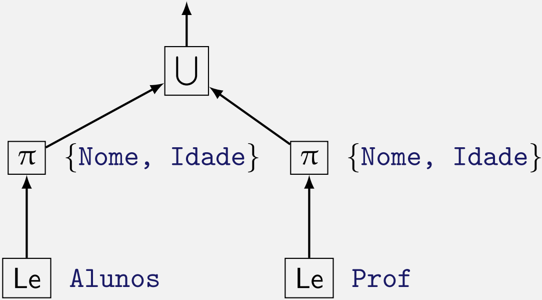
```
SELECT P1.Nome
FROM ProdL1 P1, ProdL2 P2
WHERE P1.ID = P2.ID AND
  P1.Preço < P2.Preço;
```



Operador União $\bigcup$

- Operador União: $\boxed{T_1 \bigcup T_2}$

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

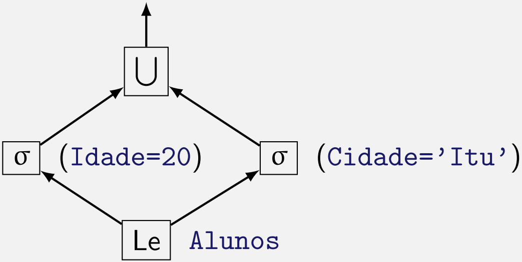


- Em SQL, `UNION` remove duplicatas, mas `UNION ALL` não remove.

Operador União $\bigcup$

- Operador União: $\boxed{T_1 \bigcup T_2}$

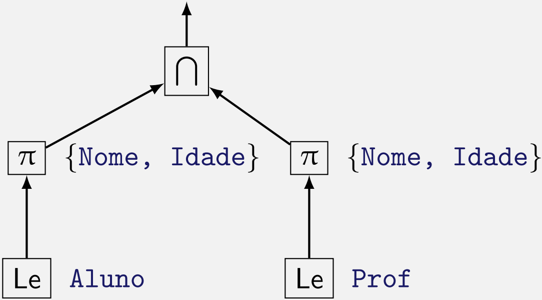
```
SELECT *
   FROM Aluno;
   WHERE Idade=20
   OR Cidade='Itu';
```



Operador Intersecção $\bigcap$

- Operador Intersecção: — $\boxed{T_1 \bigcap T_2}$

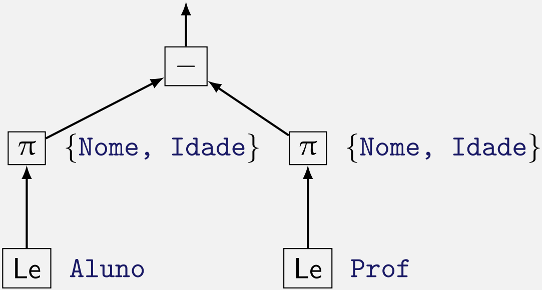
```
SELECT Nome, Idade
    FROM Aluno
  INTERSECT ALL
SELECT Nome, Idade
    FROM Prof;
```



Operador Diferença −

- Operador Diferença: — $\boxed{T_1 - T_2}$

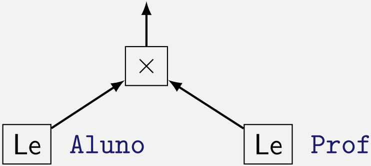
```
SELECT Nome, Idade
    FROM Aluno
  EXCEPT ALL
SELECT Nome, Idade
    FROM Prof;
```



Operador Produto Cartesiano $\times$

- Produto Cartesiano: — $\boxed{T_1 \times T_2}$

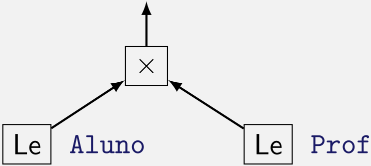
```
SELECT *
    FROM Aluno
        CROSS JOIN Prof;
```



Operador Produto Cartesiano $\times$

- Produto Cartesiano: — $\boxed{T_1 \times T_2}$

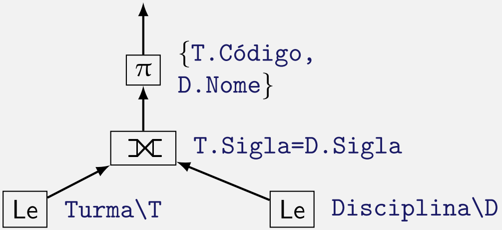
```
SELECT *
    FROM Aluno, Prof;
```



Operador Junção Externa Completa ⟗

- Operador Junção Externa Completa — $\boxed{\overset{AtrT_1=AtrT_2}{T_1 ⟗ T_2}}$

```sql
SELECT T.Código,
       D.Nome
FROM Turma T
     FULL OUTER JOIN
       Disciplina D
      ON T.Sigla = D.Sigla;
```



- “Recupere as todas turmas oferecidas e suas respectivas Disciplinas associadas, quando houver, e também mostre as Disciplinas que não tem turmas criadas.”

Projeção estendida Π

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

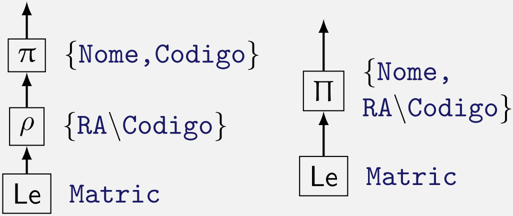
```
SELECT Nome,
     (P1+P2)/2 AS MP,
     (T1+T2+T3)/3 AS MT
   FROM Matric
   WHERE Discip='Compt';
```

$$\begin{array}{ll} \uparrow & \\ \boxed{\Pi} & \{\texttt{Nome, (P1+P2)/2}\backslash\texttt{MP, (T1+T2+T3)/3}\backslash\texttt{MT}\} \\ \uparrow & \\ \boxed{\sigma} & (\texttt{Discip='Compt'}) \\ \uparrow & \\ \boxed{\text{Le}} & \texttt{Matric} \end{array}$$

Renomear Atributos ρ

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

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



Operadores de Agregação f_{agreg}

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

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

Π Count_Nome, AVG_Nota

γ {{*}, {Count(Nome), AVG(Nota)}}

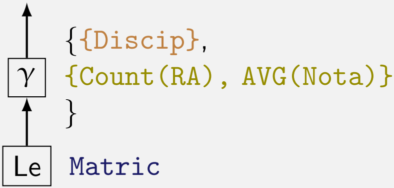
σ (Discip='Compt')

Le Matric

Operador de Agrupamento γ

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

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

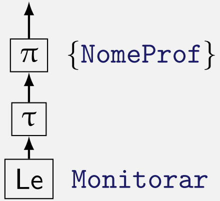


- Note que a lista de atributos do operador de agrupamento inclui dois conjuntos:
  - Os atributos do agrupamento;
  - Os atributos resultantes de funções de agregação.
- Ambos os conjuntos podem ser nulos.

Operador de Eliminação de Duplicatas τ

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

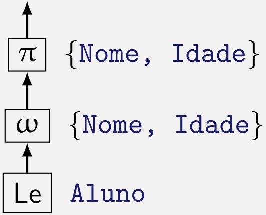
```
SELECT DISTINCT NomeProf
   FROM Monitorar;
```



Operador de Ordenação ω

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

```
SELECT Nome, Idade
   FROM Aluno
   ORDER BY Nome, Idade;
```



Roteiro

Banco de Dados
– Álgebra Relacional –

Prof. Dr. Ives Renê V. Pola
ivesr@utfpr.edu.br

Departamento Acadêmico de Informática – DAINF
UTFPR – Pato Branco DAINF
UTFPR
Pato Branco - PR

FIM