Bases de Dados

Módulo 4: Modelo Relacional – Álgebra relacional

Prof. André Bruno de Oliveira

18/07/24 23:34

Álgebra Relacional

- **Tópicos**
- **Operações que Casam Tuplas de Duas Relações**
  - Produto Cartesiano
  - Junção Natural
  - Junção Theta
- **Combinando Junção, Projeção e Seleção**
- **Operação de Renomeação**
- **Resolução em Sequência de Operações**

Álgebra Relacional - Introdução

Sobre as relações *Continente* e *Pais* definidas abaixo:

- *Regiao* (cod_reg , nome_reg)
- *Uf* (sigla_uf , nome_uf, cod_reg)

- Nota-se que *Uf* está associada à *Regiao* através do atributo em comum "cod_reg".

Regiao

cod_reg	nome_cont
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	cod_reg
RJ	Rio de Janeiro	3
SP	São Paulo	3
DF	Distrito Federal	5
PI	Piauí	2
...	...	...
GO	Goiás	5
Pará	Zimbabwe	4

Álgebra Relacional - Introdução

Considere que o usuário do BD deseja fazer a seguinte *query*.

Qual o nome da região do estado com o nome de São Paulo?

- Como resolver, se o nome da região está em *Regiao*, mas o nome do estado em *Uf*?

Regiao

cod_reg	nome_cont
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	cod_reg
RJ	Rio de Janeiro	3
SP	São Paulo	3
DF	Distrito Federal	5
PI	Piauí	2
...	...	...
GO	Goiás	5
Pará	Zimbabwe	4

Álgebra Relacional – Casamentos de Tuplas

- Para permitir a resolução de problemas deste tipo, é preciso utilizar uma operação que realize o "casamento" (match) de tuplas de uma relação com as tuplas de outra relação.
- Existem dois tipos:
  - **Produto Cartesiano**: casa as tuplas de duas relações de todas as formas possíveis.
  - **Junções**: casam de forma seletiva as tuplas de duas relações.

Álgebra Relacional – Produto cartesiano

- Combina todas as tuplas de uma relação *R* com todas de outra relação *S*.
- Denotados por: $R \times S$

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	*R*.B	*S*.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R x *S*

IBGE | ENCE Escola Nacional de Ciências Estatísticas

Álgebra Relacional – Produto cartesiano

- Denotados por: $R \times S$
- A primeira tupla de *A* combina com todas as tuplas de *B*.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	*R*.B	*S*.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R x *S*

Álgebra Relacional – Produto cartesiano

- Denotados por: $R \times S$
- A segunda tupla de *A* combina com todas as tuplas de *B*.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	*R*.B	*S*.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R x *S*

IBGE | ENCE Escola Nacional de Ciências Estatísticas

Álgebra Relacional – Produto cartesiano

- Definição Formal de Produto cartesiano (*cartesian product*) de duas relações *R* e *S*.
- Conjunto dos pares formados pela escolha do 1º elemento do par como sendo qualquer tupla de *R* e o segundo qualquer tupla de *S*. É a combinação das tupla de *R* com *S*.
- Também chamado de produto cruzado (*cross product*) ou simplesmente produto.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	*R*.B	*S*.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R x *S*

Álgebra Relacional – Produto cartesiano

- O resultado de combinar uma tupla de *R* com uma de *S* é uma tupla “ mais comprida ” (com mais atributos). O total de atributos de *R* × *S* é igual ao **número de atributos** de *R* mais o número de atributos de *S* . Neste exemplo, o total de atributos é 5 (2 + 3= 5).
- Por convenção, os componentes de *R*(operando à esquerda) precedem os componentes de S na ordem do resultado de *R* ×*S*.
- O **número de tuplas resultante** de *R*×*S* é igula ao total de tuplas de *R* multiplicado pelo total de tuplas de *S*. Neste exemplo, **o total de tuplas** é 6 (2 x 3 = 6).

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	*R*.B	*S*.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R x *S*

Álgebra Relacional – Produto cartesiano

- O esquema da relação resultante é a união dos esquemas de R e S.
- Entretanto, caso *R* e *S* possuam algum atributo com o mesmo nome , a operação de produto cartesiano realiza, **automaticamente**, a **desambiguação** no resultado.
- Isto é feito da seguinte forma: acrescentando se, como prefixo do nome do atributo, o nome da relação de onde ele é originário.
- Neste exemplo: como B é um atributo comum a ambos os esquemas, utiliza se ***R*.B** e ***S*.B** no esquema de $R \times S$.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	***R*.B**	***S*.B**	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

R x *S*

Álgebra Relacional – Junção Natural

- Combina apenas as tuplas de *R* e *S* que coincidem em quaisquer atributos que são comuns aos esquemas de *R* e *S*.
- Denotado por: R ⋈ S

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

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

R ⋈ S

Escola Nacional de Ciências Estatísticas ENCE IBGE

Álgebra Relacional – Junção Natural

- O único atributo comum entre *R* e *S* é B.
- Neste tipo de operação, para uma tupla de *R* poder casar com uma tupla de *S*, é necessário haver algum atributo com nome comum as duas relações e com mesmo valor. Neste caso o atributo de mesmo nome entre *R* e *S* é o B.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

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

$R \bowtie S$

Álgebra Relacional – Junção Natural (continuação)

- A primeira tupla de R casa com a primeira tupla de S .
- Elas compartilham o mesmo valor (valor 2) para seu atributo em comum B.
- Esse casamento gera a primeira tupla do resultado: (1,2,5,6).

A	B
1	2
3	4

Relação *R*

Casou

Não Casou

Não Casou

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	B	C	D
1	2	5	6

R ⋈ S

Álgebra Relacional – Junção Natural (continuação)

- A primeira tupla de R casa com a primeira tupla de S .
- Elas compartilham o mesmo valor (**valor 4**) para seu atributo em comum B.
- Esse casamento gera a primeira tupla do resultado: (3,4,7,8).

A	B
1	2
3	4

Relação *R*

Não Casou

Casou

Não Casou

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

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

R ⋈ S

Álgebra Relacional – Junção Natural (continuação)

- Observe que a terceira tupla de *S* não casa com nenhuma tupla de *R*.
- Não há nenhuma tupla em *R* com o valor 9 para o atributo em comum B.
- Desta forma, a terceira tupla de *S* não tem nenhum efeito no resultado final da operação de junção natural.

A	**B**
1	2
3	4

Relação *R*

B	**C**	**D**
2	5	6
4	7	8
9	10	11

Relação *S*

A	**B**	**C**	**D**
1	2	5	6
3	4	7	8

R ⋈ S

Álgebra Relacional – Junção Natural

- Se uma tupla r de *R* e uma tupla s de *S* são combinadas de forma bem sucedida em $R \bowtie S$, **a tupla combinada resultante** é chamada de ***joined tuple*** **(Tupla Unida)**.
- A *joined tuple* terá um componente para cada atributo da união dos esquemas de *R* e *S*.
- Esquemas: *R*(A,B) e *S*(B,C,D)
- União dos Esquemas: {A,B} U {B,C,D} = {A,B,C,D}

- Veja que no resultado final, o atributo **B** (o único atributo comum aos dois esquemas) aparece apenas uma vez.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

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

$R \bowtie S$

Álgebra Relacional – Junção Natural

- Definição Formal Junção Natural (natural join)
- Sejam duas relações *R* e *S*.
- Sejam A_1 , A_2 , ..., A_n **todos** os atributos que fazem parte do esquema de *R* e do esquema de *S* (atributos comuns aos dois esquemas).
- Na operação de junção natural, uma tupla r de *R* e uma tupla s de *S* serão combinadas de forma bem sucedida se e somente se r e s possuam componentes (valores) iguais para **cada um** dos atributos em comum A_1 , A_2 , ..., A_n.

Álgebra Relacional – Junção Natural

- Exemplo 2 : junção natural de relações com dois atributos em comum.

A	B	C
1	2	3
6	7	8
9	7	8

Relação *U*

B	C	D
2	3	4
2	3	5
7	8	10

Relação *V*

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

$U \bowtie V$

- A primeira tupla de *U* casa com as duas primeiras de *V*.
- A segunda e a terceira tuplas de *U* casam com a última de *V*.

Álgebra Relacional – Junção Natural

- **IMPORTANTE**: Na junção natural, para um atributo ser considerado **comum** a *R* e *S* ele precisa ter o mesmo nome nas duas relações.
- Caso um atributo represente um mesmo conceito, mas possua nomes diferentes em *R* e *S* , será necessário renomeá-lo em uma das duas relações.
- Vejamos o exemplo abaixo: O id de *Regiao* (*PK*) corresponde ao atributo cod_reg de *Uf* (*FK*). Os dois atributos correspondem ao código de região.
- Nesta situação para que seja possível aplicar a junção natural entre *Regiao* e *Uf* é preciso usar o operador de ronomeação em um dos atributos para que os nomes fiquem iguais.

Regiao

id	nome_reg
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	**cod_reg**
RJ	Rio de Janeiro	3
SP	São Paulo	3
DF	Distrito Federal	5
PI	Piauí	2
...	...	...
GO	Goiás	5
PA	Pará	4

Álgebra Relacional – Junção Theta

- A junção natural faz o casamento de tuplas usando uma condição específica: <u>A igualdade de valores dos atributos que são comuns de duas relações</u>.

  - Na prática, <u>essa é realmente condição mais frequente</u> pela qual duas relações são combinadas.
  - Isto porque, normalmente, testamos se o valor de um atributo chave estrangeira (FK – Foreign Key) em uma relação é igual ao de um atributo chave primária da outra relação (PK – Primary Key).

Álgebra Relacional – Junção Theta

- A **junção natural** faz o casamento de tuplas usando uma condição **específica**: A igualdade de valores dos atributos que são comuns de duas relações.

  - Na prática, essa é realmente condição mais frequente pela qual duas relações são combinadas.
  - Isto porque, normalmente, testamos se o valor de um atributo chave estrangeira (FK – Foreign Key) em uma relação é igual ao de um atributo chave primária da outra relação (PK – Primary Key).
  - Todavia, há situações que seja necessário combinar tuplas de duas relações usando algum outro critério.
  - Para esta propósito, existe a operação de **junção theta**.
    - Historicamente, “ theta ”, refere se a uma **condição arbitrária**.
    - Segundo a notação de (Garcia Molina et al., 2008), vamos usar a representação desta condição por *C* ao invés de θ.

Escola Nacional de Ciências Estatísticas ENCE IBGE

Álgebra Relacional – Junção Theta

- Definição Formal Junção Theta (*theta join*)
- A notação para a junção theta de duas relações R e S baseada em uma condição C é $R \bowtie_C S$.
- O resultado desta operação é construído da seguinte forma:
  I. Realizar o **produto cartesiano** entre R e S.
  II. Selecionar , deste produto, apenas as tuplas que satisfaçam a condição C.

IMPORTANTE :

I. Note que, diferente do que ocorre com a junção natural, a junção theta realiza um produto cartesiano internamente.

II. Na prática, o produto cartesiano uma operação indesejada pelos SGBDs, pois frequentemente leva ao *estouro de buffer* (memória de trabalho para armazenar resultados intermediários).

   I. De modo geral computadores com SGBD devem possuir uma configuração muito boa (vários núcleos de processamento e muita memória) para atender altas demandas de dados.

Álgebra Relacional – Junção Theta

- Exemplo : Computar $U \bowtie_{A < D} V$.

A	B	C
1	2	3
6	7	8
9	7	8

Relação *U*

B	C	D
2	3	4
2	3	5
7	8	10

Relação *V*

- Numa simulação das etapas que ocorrem internamente para resolver esta junção:
- Primeiro, faz-se o produto cartesiano $U \times V$.

Passo 1

A	*U*.B	*U*.C	*V*.B	*V*.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	2	3	4
6	7	8	2	3	5
6	7	8	7	8	10
9	7	8	2	3	4
9	7	8	2	3	5
9	7	8	7	8	10

Álgebra Relacional – Junção Natural

- Exemplo : Computar $U \bowtie_{A < D} V$.

A	B	C
1	2	3
6	7	8
9	7	8

Relação *U*

B	C	D
2	3	4
2	3	5
7	8	10

Relação *V*

- Numa simulação das etapas que ocorrem internamente para resolver esta junção:
- Segundo passo, aplicar a condição A < D para selecionar a instância (tuplas) desejada.

Passo 2

A	*U*.B	*U*.C	*V*.B	*V*.C	D
1	2	3	2	3	**4**
1	2	3	2	3	**5**
1	2	3	7	8	**10**
6	7	8	2	3	4
6	7	8	2	3	5
6	7	8	7	8	**10**
9	7	8	2	3	4
9	7	8	2	3	5
9	7	8	7	8	**10**

Álgebra Relacional – Junção Theta

- Exemplo : Computar $U \bowtie_{A < D} V$.

A	B	C
1	2	3
6	7	8
9	7	8

Relação *U*

B	C	D
2	3	4
2	3	5
7	8	10

Relação *V*

- Numa simulação das etapas que ocorrem internamente para resolver esta junção:
- Segundo passo, aplicar a condição A < D para selecionar a instância (tuplas) desejada.

Resultado final

A	***U*.B**	***U*.C**	***V*.B**	***V*.C**	**D**
1	2	3	2	3	**4**
1	2	3	2	3	**5**
1	2	3	7	8	**10**
6	7	8	7	8	**10**
9	7	8	7	8	**10**

$U \bowtie_{A < D} V$

Álgebra Relacional – Junção Theta

- Assim como ocorre no **produto cartesiano**, o esquema resultante da operação de junção theta entre duas relações *U* e *V* é uma espécie de concatenação dos esquemas de *U* e *V*.
- Veja que há situações que faz-se necessário usar o nome da relação como prefixo para diferenciar a origem de cada coluna: i) *U*.B, *U*.C ; ii) *V*.B, *V*.C. Quando o nome das colunas de *U* e *V* são iguais aplica-se esta nomenclatura no resultado final.
- Pode-se notar que a **junção theta** não elimina os atributos das relações *U* e *V*, ***diferindo da junção natural que só casa tuplas com valores iguais nos atributos***. O número de colunas da **junção theta**´usa a mesma lógica do produto cartesiano.

A	*U*.B	*U*.C	*V*.B	*V*.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

$$U \bowtie_{A < D} V$$

Álgebra Relacional – Junção Natural

- Exemplo : Computar $U \bowtie_{A < D \text{ AND } U.B \neq V.B} V$.

A	B	C
1	2	3
6	7	8
9	7	8

Relação *U*

B	C	D
2	3	4
2	3	5
7	8	10

Relação *V*

- Neste exemplo, a junção theta usa uma condição mais complexa.

Resultado final

A	*U*.B	*U*.C	*V*.B	*V*.C	D
1	2	3	7	8	**10**

- Há dois testes:

I. A < D

II. U.B ≠ V.B

- Além disso o operador AND exige que os dois testes sejam verdadeiros.
- A condição aplicada é mais refinada, resultando apenas em uma tupla.

Álgebra Relacional – Combinando operações

- A junção natural, a junção theta e o produto cartesiano podem ser combinados com outras operações da Álgebra Relacional (AR), como seleção e projeção.
- Para exemplificar, vamos usar o exemplo das relações *Uf* e *Regiao*.
    - *Regiao* (cod_reg , nome_reg,)
    - *Uf* (sigla_uf , nome_uf, cod_reg)

    - Vamos buscar a expressão AR que resolva a consulta (*query*) desejada:
        - Qual o **nome da região** do **estado brasileiro** Maranhão ?

Álgebra Relacional – Combinando operações

- Query: Qual o **nome da região** do **estado brasileiro chamado** Maranhão ?
  - *Regiao* (cod_reg , nome_reg,)
  - *Uf* (sigla_uf , nome_uf, cod_reg)

  - Para construir a expressão AR capaz de resolver a query é necessário identificar as informações:
  - **O nome da região** está contido no atributo **nome_reg** de *Regiao*.
  - **O nome do estado** encontra-se no atributo **nome_uf** da relação *Uf*.

  - Como os atributos envolvidos na consulta estão em relações diferentes, tudo indica que seja necessário utilizar uma operação que faça concatenação:
    - **Junção**
    - **Ou Produto Cartesiano**

Álgebra Relacional – Combinando operações

- Query: Qual o **nome da região** do **estado brasileiro chamado** Maranhão ?
  - *Regiao* (cod_reg , nome_reg,)
  - *Uf* (sigla_uf , nome_uf, cod_reg)

- De modo geral, costuma-se utilizar muito mais **junção natural** do que o produto cartesiano ou a junção theta.
  - Na maioria das vezes, deseja-se realizar **junção de tuplas** que possam ser “casadas” através de um *critério*.
  - Neste exemplo, *Regiao* e *Uf* estão vinculados por uma chave estrangeira (FK).
    - **cod_reg** é o atributo comum entre *Regiao* e *Uf*.

Álgebra Relacional – Combinando operações

- Query: Qual o **nome da região** do **estado brasileiro chamado** Maranhão ?
  - *Regiao* (cod_reg , nome_reg,)
  - *Uf* (sigla_uf , nome_uf, cod_reg)

- Etapas para resolver a consulta:

  I. *A* é igual realizar a junção natural entre *Regiao* e *Uf*.
  II. *B* é igual a SELECIONAR a tupla com nome_uf= 'Maranhão '.
  III. C é igual PROJETAR (B) em função do atributo **nome_reg**.

  - A expressão regular correspondente é:

$$\pi_{\text{nome\_reg}}(\sigma_{\text{nome\_uf='Maranhão'}}(Regiao \bowtie Uf\,))$$

Álgebra Relacional – Combinando operações

- Execução passo-a-passo.

- A etapa interna para encontrar a expressão regular $\pi_{nome\_reg}(\sigma_{nome\_uf=\text{'Maranhão'}}(Regiao \bowtie Uf))$ começa com:

Regiao

cod_reg	nome_reg
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	cod_reg
RJ	Rio de Janeiro	3
SP	São Paulo	3
DF	Distrito Federal	5
PI	Piauí	2
...	...	...
GO	Goiás	5
PA	Pará	4

Álgebra Relacional – Combinando operações

- Execução passo-a-passo.

- Imagine que o SGBD possui um *buffer* (memória de trabalho) que comporta um número enorme informações.
- Este *Buffer* é usado para armazenar **resultados temporários** obtidos por cada operação contida na expressão regular.
- O resultado final da *query* corresponderá ao resultado da última operação armazenada no *Buffer*.

Buffer

Álgebra Relacional – Combinando operações

- Execução passo-a-passo.

Regiao

cod_reg	nome_cont
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	cod_reg
RJ	Rio de Janeiro	3
SP	São Paulo	3
MA	Maranhão	1
RO	Rondônia	1
AC	Acre	1

$\pi_{nome\_reg}(\sigma_{nome\_uf=\text{'Maranhão'}}(Regiao \bowtie Uf))$

Buffer

Escola Nacional de Ciências Estatísticas ENCE IBGE

Álgebra Relacional – Combinando operações

- Execução passo-a-passo.

Regiao

cod_reg	nome_reg
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	cod_reg
RJ	Rio de Janeiro	3
SP	São Paulo	3
MA	Maranhão	1
RO	Rondônia	1
AC	Acre	1

$$\pi_{nome\_reg}(\sigma_{nome\_uf=\text{‘Maranhão’}}(Regiao \bowtie Uf))$$

Passo 1: resolver ($Regiao \bowtie Uf$)

Buffer

Result_temp1

cod_reg	nome_reg	sigla_uf	nome_uf
1	Norte	MA	Maranhão
1	Norte	RO	Rondônia
1	Norte	AC	Acre
3	Sudeste	RJ	Rio de Janeiro
3	Sudeste	SP	São Paulo

Escola Nacional de Ciências Estatísticas ENCE IBGE

Álgebra Relacional – Combinando operações

- Execução passo-a-passo.

$$\pi_{nome\_reg}(\sigma_{nome\_uf='Maranhão'}(Regiao \bowtie Uf))$$

Regiao

cod_reg	nome_reg
1	Norte
2	Nordeste
3	Sudeste
4	Sul
5	Centro Oeste

Uf

sigla_uf	nome_uf	cod_reg
RJ	Rio de Janeiro	3
SP	São Paulo	3
MA	Maranhão	1
RO	Rondônia	1
AC	Acre	1

Passo 2: resolver ($\sigma_{nome\_uf='Maranhão'}(Result\_temp1)$)

Buffer

Result_temp1

cod_reg	nome_reg	sigla_uf	nome_uf
1	Norte	MA	Maranhão
1	Norte	RO	Rondônia
1	Norte	AC	Acre
3	Sudeste	RJ	Rio de Janeiro
3	Sudeste	SP	São Paulo

Result_temp2

cod_reg	nome_reg	sigla_uf	nome_uf
1	Norte	MA	Maranhão

Resposta

nome_reg
Norte

Álgebra Relacional – Renomeação

- O operador de renomeação (renaming), denotado por ρ, é utilizado para renomear relações e atributos.
  - Exemplo 1: Renomear a relação e os atributos:
    - $\rho_{s(A1,A2,...,An)}(R)$
      - Nesta expressão, *R* é o nome da relação original.
      - *S* é a relação resultante após a troca de nome. *S* contém as mesmas tuplas de *R*.
      - Além disso, a expressão troca os nomes dos atributos de *R* para $A_1, A_2, ..., A_n$. Assim, a relação resultante *S* tem os nomes diferentes da relação original.
  - Exemplo 2: Renomear apenas a relação.
    - $\rho_s(R)$
      - Neste caso, o nome da relação muda para *S* e os nomes dos atributos ficam inalterados.

Álgebra Relacional – Renomeação

- Vejamos um exemplo de renomeação combinado com um produto cartesiano.
  - Ambas as relações de $R \times S$, contém o atributo B.
  - Considere a seguinte expressão: $R \times \rho_{s(K,C,D)}(S)$.
    - É efetuado um produto cartesiano comum, só que desta vez o atributo S.B muda de nome para K.

A	B
1	2
3	4

Relação *R*

B	C	D
2	5	6
4	7	8
9	10	11

Relação *S*

A	*R*.B	*K*	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

$R \times \rho_{s(K,C,D)}(S)$

Álgebra Relacional – Sequência de Operações

- Muito frequentemente, a resolução de uma consulta envolve a combinação de diversas operações.
- Para facilitar a resolução, nós intuitivamente costumamos elaborar uma sequência de passos capaz de solucionar a consulta.

- Ex.: Qual o **nome da região** do **estado brasileiro** Maranhão ?
  - *Regiao* (cod_reg , nome_reg,)
  - *Uf* (sigla_uf , nome_uf, cod_reg)
- **Etapas para resolver a consulta:**
  – *A* é igual realizar a junção natural entre *Regiao* e *Uf*.
  – *B* é igual a SELECIONAR a tupla com nome_uf= 'Maranhão '.
  – C é igual PROJETAR (B) em função do atributo **nome_reg**.

Álgebra Relacional – Sequência de Operações

- A Álgebra Relacional permite com que expressemos cada passo como um **comando de atribuição**.
  - Essencialmente, a resolução de uma consulta pode ser expressa como uma sequência de comandos de atribuição.
  - Cada comando gera uma nova relação no buffer, na forma de um resultado temporário, conforme visto anteriormente.
  - Um comando de atribuição tem o seguinte formato:
    - nome := operação
    - Onde **nome** representa o nome da relação produzida pela operação indica à direita do símbolo de atribuição ":=".
  - Conforme visto nos slides anteriores, a relação gerada pelo último comando corresponderá ao resultado final da consulta.
    - Para padronizar, a chamaremos sempre de **Resposta** em nossos exemplos.

Álgebra Relacional – Sequência de Operações

- A Álgebra Relacional permite com que expressemos cada passo como um **comando de atribuição**.
  - Essencialmente, a resolução de uma consulta pode ser expressa como uma sequência de comandos de atribuição.
  - Cada comando gera uma nova relação no buffer, na forma de um resultado temporário, conforme visto anteriormente.
  - Um comando de atribuição tem o seguinte formato:
    - nome := operação
    - Onde **nome** representa o nome da relação produzida pela operação indica à direita do símbolo de atribuição ":=".
  - Conforme visto nos slides anteriores, a relação gerada pelo último comando corresponderá ao resultado final da consulta.
    - Para padronizar, podemos chamar o resultado de final sempre de **Resposta** em nossos exemplos.

Álgebra Relacional – Sequência de Operações

- Exemplo 1: Qual o **nome da região** do **estado brasileiro** Maranhão ?
  - *Regiao* (cod_reg , nome_reg,)
  - *Uf* (sigla_uf , nome_uf, cod_reg)

  - RT1 := *Regiao* ⋈ *Uf*
  - RT2 : = $\sigma_{\text{nome\_uf='Maranhão'}}$(RT1)
  - Resposta := $\pi_{\text{nome\_reg}}$(RT2)

  - Representação como expressão única (Notação *in-Line*)

$$\pi_{\text{nome\_reg}}(\sigma_{\text{nome\_uf='Maranhão'}}(Regiao \bowtie Uf))$$

Álgebra Relacional – Sequência de Operações

- Exemplo 2: Quais são os carros nacionais e seus respectivos anos de fabricação com pelos menos 100.000 de km rodados?
  - *Carro* (carro , ano, cor, pais, km, avaliacao)

  Resolução em sequência de operações

  - Temp := $\sigma_{km \geq 100.000 \text{ AND } pais=\text{'BR'}}(Carro)$
  - Resposta := $\pi_{carro}(Temp)$

  - Representação como expressão única (Notação *in-Line*)

$$\pi_{carro,ano}(\sigma_{km \geq 100.000 \text{ AND } pais=\text{'BR'}}(Carro))$$

Escola Nacional de Ciências Estatísticas ENCE IBGE

Álgebra Relacional – Sequência de Operações

- Exemplo 3: Quais são os nomes dos países que lançaram carro antes de 1980?
  - *Carro_lançado* (carro, ano, modelo, cor, pais)
  - *Pais* (sigla, nome)
  - Dado que o atributo pais da relação *Carro_lançado* contem a sigla do pais de lançamento do carro.

  Resolução em sequência de operações

  T1985 := $\sigma_{ano<1980}(\textit{Carro\_lançado})$

  Temp1 (carro, ano, modelo, cor, **sigla**) := T1985

  (aqui T1985 é atribuído a Temp1. Temp1 é uma relação temporária obtida previamente pela renomeação do atributo **pais** para a **sigla** na relação *Carro_lançado*)

  Temp2 := $Pais \bowtie Temp1$

  Resposta := $\pi_{nome}(Temp2)$

  - Representação como expressão única (Notação *in-Line*)

  $$\pi_{nome}\ (Pais \bowtie (\rho_{s(\underline{carro},\underline{ano},\underline{modelo},\ cor,\ sigla)}(\sigma_{ano<1980}(\textit{Carro\_lançado}))))$$

Escola Nacional de Ciências Estatísticas | ENCE | IBGE

Álgebra Relacional – Quadro resumo

Operação	Finalidade	Notação
União	Produz uma relação que inclui todas as tuplas que estão em *R* e *S*. As tuplas duplicadas são eliminadas. É uma operação comutativa.	$R \cup S$
Interseção	Produz uma relação que inclui todas as tuplas que comuns entre *R* e *S*. É uma operação comutativa.	$R \cap S$
Diferença	Produz uma relação com todas as tuplas contidas em *R* e não em *S*. **Não** é uma operação comutativa.	$R - S$
Seleção	Seleciona todas as tuplas que satisfazem a condição de seleção de uma relação *R*.	$\sigma_C(R)$
Projeção	Produz uma nova relação com apenas alguns dos atributos de R, e remove tuplas duplicadas.	$\pi_C(R)$
Junção Natural	Produz todas as combinações de tuplas de *R* e *S* que satisfazem uma condição de junção apenas com comparações de igualdade. Os atributos de *S* usados na comparação não são incluídos.	$R \bowtie S$
Junção Theta	Produz todas as combinações de tuplas de R1 e R2 que satisfazem a condição de junção.	$R \bowtie_C S$
Produto cartesiano	Produz uma relação que tem os atributos de *R* e *S* com todas as possíveis combinações de tuplas. É uma operação comutativa.	$R \times S$
Renomeação	Produz uma relação contendo todas as tuplas de *R* trocando o nome da relação e ou de cada atributo.	$\rho_{s(a_1,\ldots,a_n)}(S)$

Álgebra Relacional – Exercício proposto

- Considere uma parte das instâncias das relações *Projeto* e *Funcionario* apresentadas a seguir. Em *Funcionario* , o atributo “ID” é chave estrangeira referenciando o campo “ID” de Projeto.

Projeto

id	siglaProjeto
1	MUNIC
2	ESTADIC
3	PNAD
4	PNSB

Funcionario

matricula	nomeFunc	id
10	Arlekina	4
20	Eva	2
30	Brad	1
40	Drake	3
50	Vin	1
60	Nina	2
....	...	....

- Elabore as expressões de Álgebra Relacional para responder as seguintes perguntas:
- a) Recuperar a sigla dos projetos em que os funcionários de matrícula 50 e 90 estão alocados.
- b) Recuperar a sigla do projeto, as matrículas e os nomes dos funcionários alocados ao projeto de id igual a 4.
- c) Listar o ID de todos os projetos que possua somente 1 empregado alocado.
- d) Desenhar a query tree referente a solução (b).

Gabarito Aula 8 – Parte 2

a) Recuperar as **siglas dos projetos** em que os funcionários de matrícula 50 e 90 estão alocados.

Projeto (id, siglaprojeto)

Funcionario (*matricula*, nomeFunc, id)

Temp1 := $\sigma_{\text{matricula=50 OR matricula=90}}(\textit{Funcionario})$

Temp2 := $\textit{Projeto} \bowtie \textit{Temp1}$

Temp3 := $\pi_{\text{siglaProjeto}}(\textit{Temp2})$

$$\pi_{\text{siglaProjeto}}(\textit{Projeto} \bowtie (\sigma_{\text{matricula=50 OR matricula=90}}(\textit{Funcionario})))$$

Escola Nacional de Ciências Estatísticas ENCE | IBGE

b) Recuperar a **sigla do projeto**, as **matrículas** e os **nomes dos funcionários** alocados ao projeto de id igual a 4.

Projeto (id, siglaprojeto)

Funcionario (*matricula*, nomeFunc, id)

Temp1 := $\sigma_{id=4}(\textit{Projeto})$

Temp2 := $\textit{Funcionario} \bowtie \textit{Temp1}$

Temp3 := $\pi_{siglaProjeto,matricula,nomeFunc}(\textit{Temp2})$

$$\pi_{siglaProjeto,matricula,nomeFunc}(\textit{Funcionario} \bowtie (\sigma_{id=4}(\textit{Projeto})))$$

Escola Nacional de Ciências Estatísticas ENCE IBGE

c) Listar o ID de todos os projetos que possuam menos de 2 empregados alocados.

Projeto (id, siglaprojeto)

Funcionario (*matricula*, nomeFunc, id)

A1 := $\pi_{\text{matricula,id}}(\textit{Funcionario})$

A2 := $\pi_{\text{matricula,id}}(\textit{Funcionario})$

A3:= $A1 \times A2$

(agora vamos selecionar as tuplas que ocorrem mais de uma vez)

A4 := $\sigma_{\text{A1.id=A2.id AND A1.matricula} \neq \text{A2.matricula}}(A3)$

Raciocínio detalhando no próximo slide

Raciocínio

A1 × A2

A1

matricula	id
10	4
20	2
30	1
40	3
50	1
60	2
....	....

A2

matricula	id
10	4
20	2
30	1
40	3
50	1
60	2
....	....

Raciocínio

$A1 \times A2$

A1

matricula	id
10	4
20	2
30	1
40	3
50	1
60	2
....	....

A2

matricula	id
10	4
20	2
30	1
40	3
50	1
60	2
....	....

A3

A1.matricula	A1.id	A2.matricula	A2.id
...	...	...	...
20	2	10	4
~~20~~	~~2~~	~~20~~	~~2~~
20	2	30	1
20	2	40	3
20	2	50	1
20	2	60	2
....	....	....	....

- Como exemplo, vamos usar o projeto de **id** igual 2. De acordo com a condição, somente é selecionada a tupla do mesmo projeto com matrícula diferente. Assim, restam somente as tuplas com matrículas diferentes para este mesmo **id**.
- Os casos em que o id contém somente uma matrícula ficam de fora do resultado temporário de A3. Digamos que o projeto de id igual 3 tivesse somente um funcionário. Aplicando o mesmo raciocínio, este projeto ficaria de fora.

$$\sigma_{\text{A1.id=A2.id AND A1.matricula} \neq \text{A2.matricula}}(A3)$$

c) Listar o ID de todos os projetos que possuam menos de 2 empregados alocados.

Projeto (id, siglaprojeto)

Funcionario (*matricula*, nomeFunc, id)

$A1 := \pi_{matricula,id}\, Funcionario$

$A2 := \pi_{matricula,id} Funcionario$

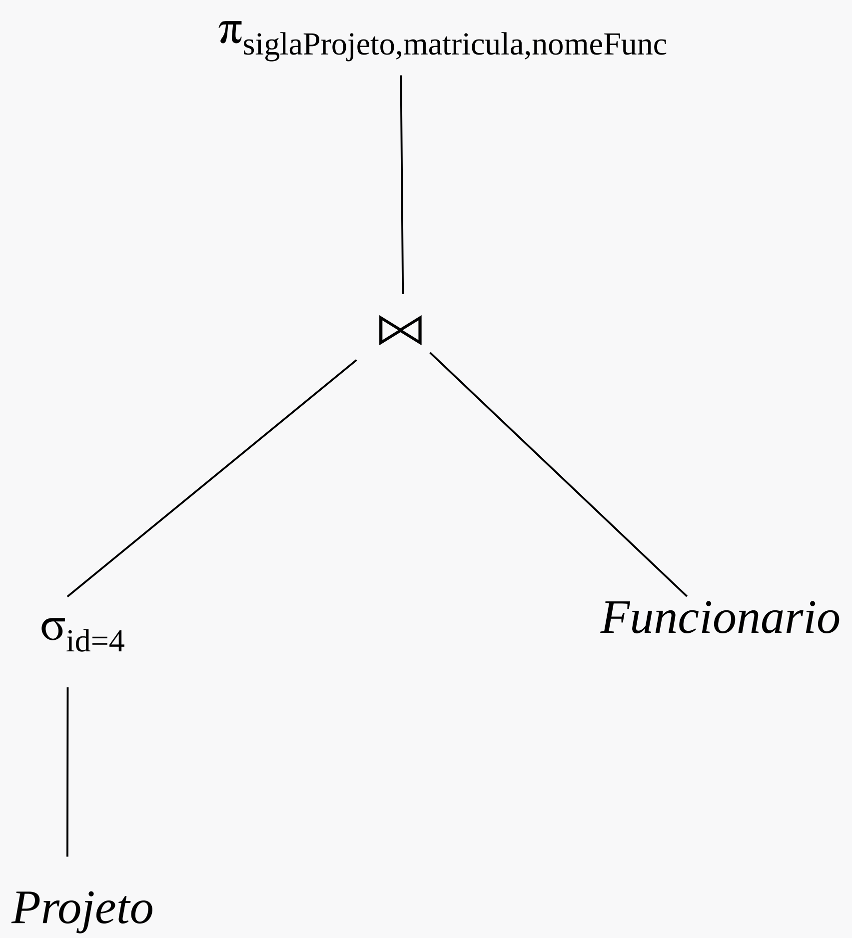
$A3 := A1 \times A2$

$Temp4 := \sigma_{A1.id=A2.id \text{ AND } A1.matricula \neq A2.matricula}(A3)$

$Temp5 := Resultado := \pi_{id}\,(funcionário) - \pi_{id}\,(Temp4)$

d) Desenhar a query tree referente a solução (b).

$\pi_{siglaProjeto,matricula,nomeFunc}\ (\mathit{Funcionario} \bowtie (\sigma_{id=4}(\mathit{Projeto})))$



Obrigado