



**Curso:** Análise e Desenvolvimento de Sistemas

**Disciplina:** Banco de Dados I

**Professor:** Paulo Giovani

**Aluna:** Matheus de Jesus Antunes. Matrícula: CJ302475x

## **Álgebra Relacional e Cálculo Relacional: conceito, funções e operações**

### **Introdução**

O presente trabalho acadêmico se apresenta como avaliação parcial da disciplina Banco de Dados I, sendo elaborado com o objetivo de contribuir para a compreensão do conteúdo programático da disciplina Banco de Dados I sobre Álgebra Relacional e Cálculo Relacional.

Sendo assim, de forma a proporcionar o conhecimento e compreensão sobre o conteúdo de Álgebra Relacional e Cálculo Relacional, o mesmo está dividido em três partes, buscando abordar aspectos conceituais, funções, operações, bem como exemplos de consultas em Álgebra Relacional e sua equivalência em Cálculo Relacional.

### **1 - Conceituado Álgebra Relacional**

Conforme já apresentado durante as aulas da Disciplina Banco de Dados I, um modelo de dados inclui um conjunto de operações para manipular o “Banco de Dados” e conceitões de “Modelo de Dados” para definição das restrições e estrutura do Banco de Dados.

Sendo assim, para abordarmos a “Álgebra Relacional”, primeiramente se faz necessário considerarmos a estrutura dos “Bancos de Dados Relacionais”, que segundo Silberschatz, Korth e Sudarshan (2006), consiste em uma coleção de tabelas, cada uma com um nome único atribuído.

O conjunto básico de operações para o modelo relacional é a “Álgebra Relacional”, considerada frequentemente como uma parte do modelo relacional de dados (ELMASRI e NAVATHE, 2005).

Com base nisso, parto do princípio que a “Álgebra Relacional”, se apresenta como um conjunto básico de operações que nos permite manipular relações no modelo relacional.

## **1.1 - Funções da Álgebra Relacional**

Para Date (2004) a “Álgebra Relacional” é uma coleção de operadores que tomam relações como seus operandos e retornam uma relação como seu resultado. Silberschatz, Korth e Sudarshan (2006), expõem que a maioria dos dados relacionais comerciais oferece uma linguagem de consulta que inclui elementos procedural e não procedural, no qual a “Álgebra Relacional” é procedural.

O objetivo fundamental da “Álgebra Relacional” é permitir a escrita de expressões relacionais, que servem como uma representação simbólica de alto nível da intenção do usuário (DATE, 2004).

Portanto, entende-se que ao utilizarmos a “Álgebra Relacional”, o usuário dá as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para calcular o resultado desejado.

## **1.2 – Operações de Álgebra Relacional**

As operações fundamentais da “Álgebra Relacional” nos permite fornecer uma definição completa de uma expressão na álgebra relacional (SILBERSCHATZ, KORTH e SUDARSHAN, 2006).

As operações de “Álgebra Relacional” podem ser divididas em dois grupos basicamente, sendo um que inclui um conjunto de operações da teoria de conjunto matemática e o outro desenvolvidas especificamente para os bancos de dados relacionais (ELMASRI e NAVATHE, 2005).

Elmasri e Navathe (2005) apresentam como conjunto de operações matemáticas a UNIÃO (UNION) – ( $\cup$ ); INTERSEÇÃO (INTERSECTION) – ( $\cap$ ); DIFERENÇA DE CONJUNTO (SET DIFFERENCE) - ( $-$ ) e PRODUTO CARTESIANO (CROSS PRODUCT) - ( $\times$ ).

No que se refere as operações desenvolvidas para banco de dados Elmasri e Navathe (2005) expõe que estás incluem a SELEÇÃO (SELECT) – ( $\sigma$ ); PROJEÇÃO (PROJECT) – ( $\pi$ ) e JUNÇÃO (JOIN) – ( $\bowtie$ ).

Segundo Elmasri e Navathe (2005), as operações SELEÇÃO e PROJEÇÃO são unárias. Seguem as descrições a seguir:

A SELEÇÃO (SELECT) é usada para escolher um subconjunto de tuplas de uma relação que satisfaça uma condição de seleção (WILL, 2022).

Segundo Will (2022), a operação SELEÇÃO pode ser considerada como um filtro que mantém apenas as tuplas que satisfaçam uma condição qualificadora, se restringindo as tuplas em uma relação para apenas aquelas que satisfaçam a condição, sendo descartadas aquelas que não satisfazem a condição.

O símbolo  $\sigma$  (sigma) é usado para indicar o operador SELEÇÃO e a condição de seleção é uma expressão booleana (condição) especificadas nos atributos da relação R. Em geral, a operação SELEÇÃO é indicada por  $\sigma_{\langle \text{condicao\_selecao} \rangle}(R)$  (WILL, 2022). Já a PROJEÇÃO (PROJECT), segue sua operação selecionando certas colunas da tabela enquanto descarta outras, no qual se estivermos interessados apenas em certos atributos de uma relação, usamos a operação PROJEÇÃO para projetar a relação apenas por esses atributos (WILL, 2022).

Sendo assim, conforme ressaltado por (Will, 2022), o resultado da operação PROJEÇÃO pode ser visualizado como uma partição vertical da relação em duas relações: uma tem as colunas (atributos) necessários e contém o resultado da operação, e a outra contém as colunas descartadas.

A forma geral da operação PROJEÇÃO é  $\pi_{\langle \text{lista\_atributos} \rangle}(R)$ , onde  $\pi$  ( $\pi$ ) é o símbolo usado para representar a operação PROJEÇÃO, e  $\langle \text{lista\_atributos} \rangle$  é a sublista desejada de atributos da relação R (WILL, 2022).

O R, em geral, é uma expressão da álgebra relacional cujo resultado é uma relação, que no caso mais simples é apenas o nome de uma relação do banco de dados (WILL, 2022). O resultado da operação PROJEÇÃO tem apenas os atributos especificados em  $\langle \text{lista\_atributos} \rangle$  na mesma ordem em que eles aparecem na lista. Logo, seu grau é igual ao número de atributos em  $\langle \text{lista\_atributos} \rangle$  (WILL, 2022).

Seguindo a apresentação das operações relacionais desenvolvidas para Banco de Dados, apresentam-se também as operações Binárias, que são a de JUNÇÃO (JOIN) e a Operação de DIVISÃO (DIVISION) (WILL, 2022).

Segundo Elmasri e Navathe (2005), a operação JUNÇÃO é indicada pelo símbolo  $\bowtie$  é usada para combinar tuplas relacionadas de duas relações em uma única tupla “maior”, permitindo combinar tuplas de duas relações que obedecem a uma condição de junção.

A operação de junção é representada pela expressão  $R \text{ COND } S$ , onde  $\bowtie$  é o operador de junção, COND é a condição de junção e R e S são duas relações quaisquer (MELO, 2019).

A condição COND é da forma  $A \theta B$ , onde  $\theta$  é um operador de comparação  $\{=, <, >, \leq, \geq, \neq\}$ , A; é um atributo de R e B é um atributo de S. A condição COND também pode ser composta por várias cláusulas ligadas pelo operador lógico AND (MELO, 2019). Esta operação de JUNÇÃO é muito importante para qualquer banco de dados relacional com mais de uma relação única, porque nos permite processar relacionamentos entre as relações, no qual em seu uso mais comum envolve as condições apenas em comparação a igualdade (ELMASRI e NAVATHE, 2005).

Com base na classificação de JUNÇÃO, existem variações que são a JUNÇÃO NATURAL (NATURAL JOIN) e a EQUIJUNÇÃO (EQUIJOIN) (ELMASRI e NAVATHE, 2005).

As operações de JUNÇÃO NATURAL como uma operação binária, nos permite combinar certas seleções e um produto cartesiano dentro de uma operação formando um produto cartesiano de seus dois argumentos. Esta operação promove uma seleção obedecendo à equivalência dos atributos que aparecem em ambos os esquemas de relação e, finalmente, removem os atributos em duplicidade. Sua definição padrão exige que os dois atributos de junção tenham o mesmo nome em ambas relações (ELMASRI e NAVATHE, 2005).

Já a EQUIJUNÇÃO (EQUIJOIN) é uma junção em que o único operador de comparação for o  $=$ , no qual no seu resultado sempre teremos um ou mais pares de atributos com valores idênticos em cada tupla (ELMASRI e NAVATHE, 2005).

Conforme já enunciado, também temos a operação Binária DIVISÃO (DIVISION) que é indicada pelo símbolo  $\div$ , permite obter os valores de uma relação que estão combinados com todas as tuplas de outra relação. Essa operação é útil para um tipo especial de consulta (ELMASRI e NAVATHE, 2005).

A operação de divisão é representada pela expressão  $R \div S$ , onde  $\div$  é o operador da divisão e R e S são duas relações em que os atributos de S são um subconjunto dos atributos de R (MELO, 2019).

De acordo com Elmasri e Navathe (2005) divisão é útil para um tipo especial de consulta que, as vezes, ocorre em aplicações de banco de dados. Trata-se de uma operação binária que exhibe todos os elementos da primeira tabela que se relacionam com todos os elementos da segunda tabela.

É importante ressaltar que considera-se Unárias, as operações que envolvem apenas uma tabela e Binárias: envolvem duas tabelas.

## **2- Cálculo Relacional: Conceito e função**

O Cálculo Relacional apresenta-se como uma linguagem de consulta formal. Utilizando-se de uma expressão declarativa pode-se especificar uma consulta. Uma expressão de cálculo permite a descrição da consulta desejada sem especificar os procedimentos para obtenção dessas informações, ou seja, é não-procedural (ELMASRI e NAVATHE, 2005).

Se tratando de uma linguagem de consulta se escreve uma expressão declarativa (o que fazer ao invés de como fazer).

Conforme apresentado por Elmasri e Navathe (2005) existem dois tipos de cálculos: Cálculo Relacional de Tuplas (CRT) Cálculo Relacional de Domínio (CRD).

Para Elmasri e Navathe (2005), o Cálculo Relacional de Tuplas (CRT) é baseado na especificação de um número de variáveis de tuplas. Uma consulta em CRT é especificada da seguinte forma:

$\{t \mid \text{COND}(t)\}$ , onde  $t$  é uma variável e  $\text{COND}(t)$  é uma expressão condicional. Já o Cálculo Relacional de Domínio (CRD) é baseado na especificação de variáveis sobre o domínio dos atributos, onde cada variável pode tomar como valor os valores de um determinado domínio (ELMASRI E NAVATHE, 2005).

Uma fórmula de CRD pode ser formada por condições da forma  $R(v_1, v_2, \dots, v_n)$  ou  $v; \theta v$  ou  $v; \theta \text{VAL}$ , onde  $R$  é uma relação, cada  $v$  é uma variável que representa valores do domínio do atributo  $A$ ; de  $R$ ,  $\theta$  é um operador de comparação  $\{=, <, >, \leq, \geq, \neq\}$  e  $\text{VAL} \in \text{dom}(A;)$  (MELO, 2019).

Uma fórmula de CRD também pode ser composta por várias fórmulas ligadas pelos operadores lógicos AND, OR ou NOT ou usar quantificadores universal e existencial  $F1 \text{ AND } F2$  ou  $F1 \text{ OR } F2$  ou  $\text{NOT } F1$ , onde  $F1$  e  $F2$  são fórmulas,  $\forall \exists$  são os quantificadores universal e existencial e  $v$  é uma variável que representa valores de um domínio (MELO, 2019).

O Cálculo Relacional de Domínio se difere do cálculo das tuplas pelos tipos de variáveis, pois aqui elas abrangem os valores únicos dos domínios dos atributos. Ou seja, o cálculo relacional de tupla usa as variáveis de tuplas (linhas) das relações e o cálculo relacional de domínio usa as variáveis de domínio que engloba os domínios (coluna das relações) (ELMASRI E NAVATHE, 2005).

### **3 – Exemplos de Consultas em Álgebra Relacional e seu equivalente em Cálculo Relacional**

Partindo do princípio conceitual e da apresentação de suas funções e operações, é possível identificar que enquanto a Álgebra Relacional define um conjunto de operações para o modelo relacional, o Cálculo Relacional provê uma notação declarativa de nível superior para a especificação de consultas relacionais (ELMASRI E NAVATHE, 2005).

Conforme ressaltado por Elmasri e Navathe (2005), ao apresentar distinções uma expressão de cálculo relacional cria uma nova relação, que é especificada em termos de variáveis que abrangem as linhas das relações armazenadas no banco de dados (em cálculo de tuplas) ou as colunas das relações armazenadas (em cálculo de domínio).

Para Elmasri e Navathe (2005) a principal característica de distinção entre Álgebra Relacional e o Cálculo Relacional é que em uma expressão de cálculo relacional, não há ordem nas operações para especificar como recuperar o resultado de uma consulta, especifica apenas qual informação o resultado deveria conter.

Com base nisso, embora possuam distinções, percebe-se que tanto o cálculo relacional quanto a álgebra relacional são duas linguagens importantes para a manipulação de dados em bancos de dados relacionais.

Além disso, existe uma equivalência matemática entre Álgebra Relacional e Cálculo Relacional, o que favorece na possibilidade de expressar qualquer consulta em uma linguagem também na outra. Isso significa que qualquer consulta formulada em álgebra relacional pode ser expressa no cálculo relacional e vice-versa.

Seguem abaixo exemplos de operações em Álgebra Relacional e Cálculo Relacional (Spaceprogrammer):

Seleção:  $\sigma$

A seleção, como o próprio nome já diz, seleciona linhas, tuplas por meio de uma determinada condição.

Ex:  $\sigma_{\text{nome}=\text{"Daniel"}}(\text{cliente})$

nome	cpf
Daniel	1234
Carlos	2233
Diego	9908
Robert	3276

=

nome	cpf
Daniel	1234

Projeção:  $\pi$

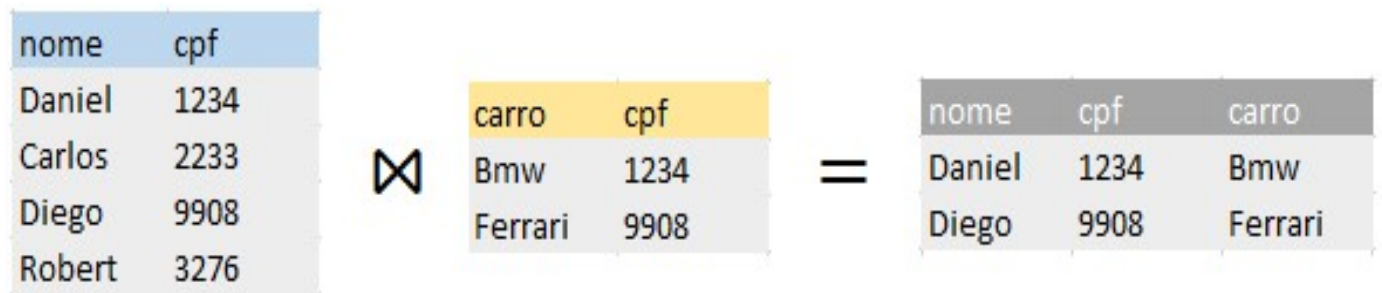
A projeção é utilizada quando existe a necessidade de pegar somente colunas de interesse em uma relação, e não trabalhar com todas as colunas dessa relação. Ex:  $\pi_{\text{nome}}(\text{cliente})$

nome	cpf
Daniel	1234
Carlos	2233
Diego	9908
Robert	3276

=

nome
Daniel
Carlos
Diego
Robert

Junção:  $\bowtie$   
(cliente)  $\bowtie$  (veículo)



Utilizando a linguagem de Cálculo Relacional de Tuplas, vamos considerar uma tabela chamada "Vendas" com os atributos "Produto", "Quantidade" e "Preço".

Desejamos selecionar todas as tuplas em que a quantidade de vendas seja superior a 100 unidades. Podemos expressar essa consulta usando o cálculo de tuplas da seguinte maneira:

{Produto, Quantidade, Preço |  $\exists$  quantidade.(Quantidade > 100)}

Essa consulta seleciona todas as tuplas da tabela de vendas em que a quantidade é maior que 100 unidades, retornando o produto, a quantidade e o preço correspondentes.

Utilizando Cálculo Relacional de domínio se tivermos uma tabela de funcionários com os atributos "nome" e "salário", poderíamos formular uma consulta para encontrar os funcionários cujo salário seja maior que \$5000 da seguinte maneira:

{nome | salario > 5000}

O resultado dessa consulta seria uma tabela contendo os nomes dos funcionários que atendem à condição especificada.



## **Conclusão**

Partindo das considerações acerca da linguagem de Álgebra Relacional e Cálculo Relacional, tira-se a conclusão de que as operações de Álgebra Relacional permitem a um usuário especificar as solicitações básicas usadas dentro das linguagens de consulta relacional.

No que tange, ao Cálculo Relacional, este oferece uma linguagem declarativa, no qual uma expressão gera uma nova relação, sendo muito importante por conta da sua base lógica matemática consistente e porque a SQL para os SGBDRs, assuntos que ainda serão abordados na disciplina, possui muitos de seus fundamentos no Cálculo Relacional de Tupla.

Contudo, é importante ressaltar que a explicação e compreensão acerca do conteúdo de Álgebra Relacional e Cálculo Relacional não se esgota ao que foi apresentado neste trabalho acadêmico, pois a mesma exige a realização de prática de exercícios, bem como análise e sistematização do mesmo.

## **Referências**

Aprendendo as principais operações da Álgebra Relacional. In: <https://spaceprogrammer.com>  
Acesso em: 05 de setembro de 2023.

DATE, C. J. Introdução a Sistemas de Bancos de Dados. Tradução de Daniel Vieira,  
Rio de Janeiro, Elsevier, 2004.

ELMASRI, R.; NAVATHE, S. B. Sistemas de Bancos de Dados. São Paulo, Pearson  
Addison Wesley, 2005.

MELO, Tiago Eugenio de. Álgebra e Cálculo Relacional. Escola Superior de Tecnologia  
Universidade do Estado do Amazonas, 7 de Junho de 2019.

SILBERSCHATZ, A.; KORF, H. F.; SUDARSHAN, S. Sistemas de Banco de Dados,  
1a. Ed. Rio de Janeiro, Elsevier, 2006.

WILL, Newton Carlos. Apostila Álgebra Relacional. Universidade Tecnológica Federal do  
Paraná, 2022.