**[Resumo sobre algoritmos de operações da álgebra relacional](https://eadsalvador.ifba.edu.br/moodle/mod/assign/view.php?id=34615)**

****

**Dupla:**   
Gabriel Ferreira de Jesus

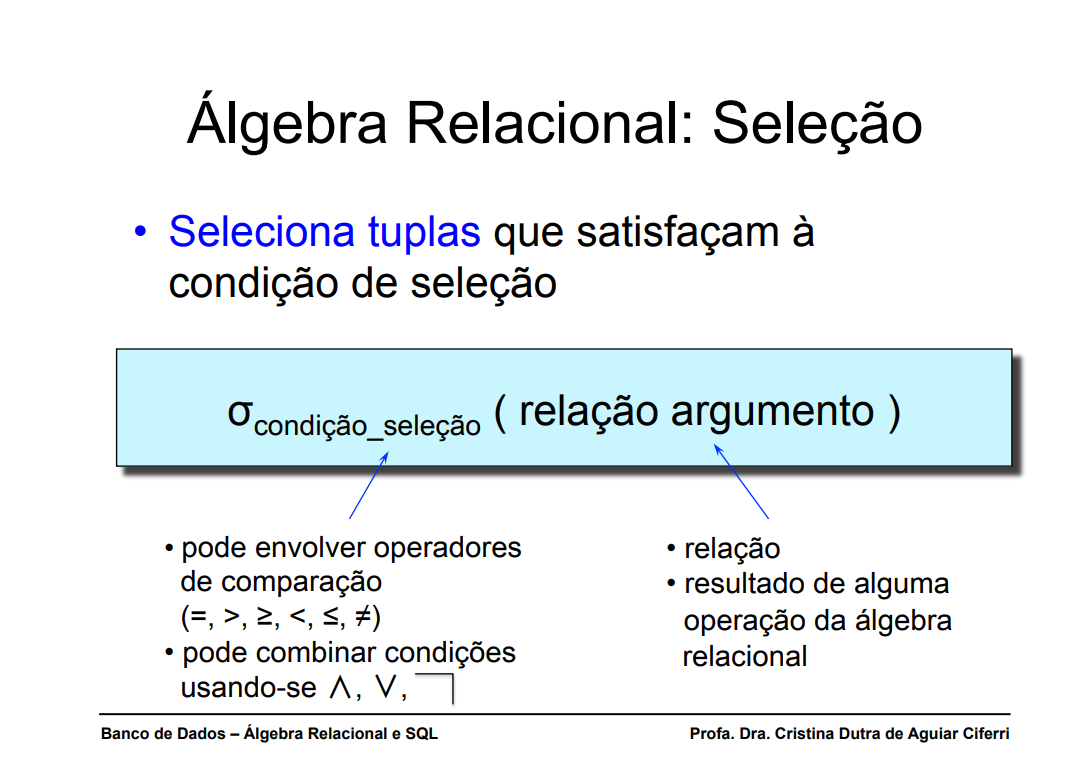
Rafael Oliveira Ledo

**Resumo**

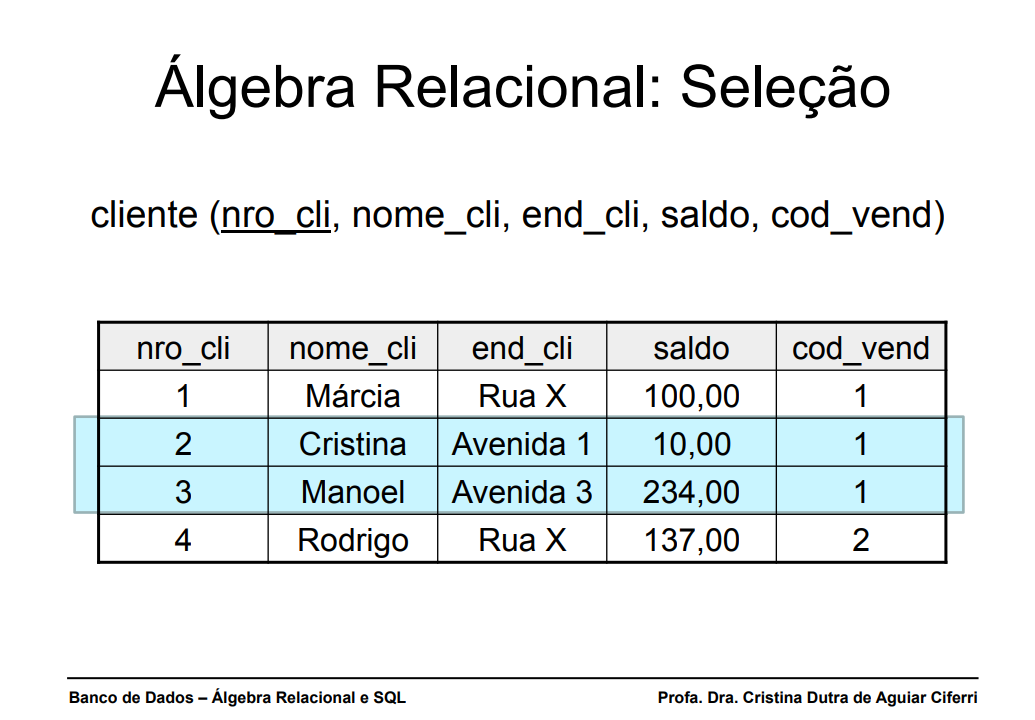
A álgebra relacional é uma ferramenta poderosa para entender o funcionamento de um banco de dados desde a base, com princípios e conhecimentos fundamentais para um analista de dados. A linguagem SQL usada na maior parte dos bancos de dados, é fundamentada em álgebra relacional que é uma derivação descendente da lógica de primeira ordem e da álgebra de conjuntos. Ela ficou conhecida em 1970 com a publicação do modelo relacional de dados de Edgar Frank Cood, que propôs a mesma como base das linguagens de consulta de banco de dados. A álgebra relacional pode ser entendida como uma coleção de operações de alto nível, que são elas: 1-restrição(seleção), 2-projeção, 3-produto, 4-união, 5-diferença, 6-interseção, 7-junção, 8-divisão, 9-agregação. Dessas 9 operações apenas as 5 primeiras respectivamente são primitivas as outras 4 são consideradas operações de conjuntos.   
Sua fundamentação formal baseada na lógica faz com que sua otimização seja possível e as melhorias possam ser contínuas a partir de uma análise detalhada das consultas. Abordaremos cada uma delas nesse resumo logo a seguir.

**1.Seleção - σ**

A operação de Seleção é uma das mais usadas sem sombra de dúvidas pelos usuários de SQL pois é a partir dela que selecionamos quais são os resultados queremos que sejam processados a partir da condição que iremos escolher.



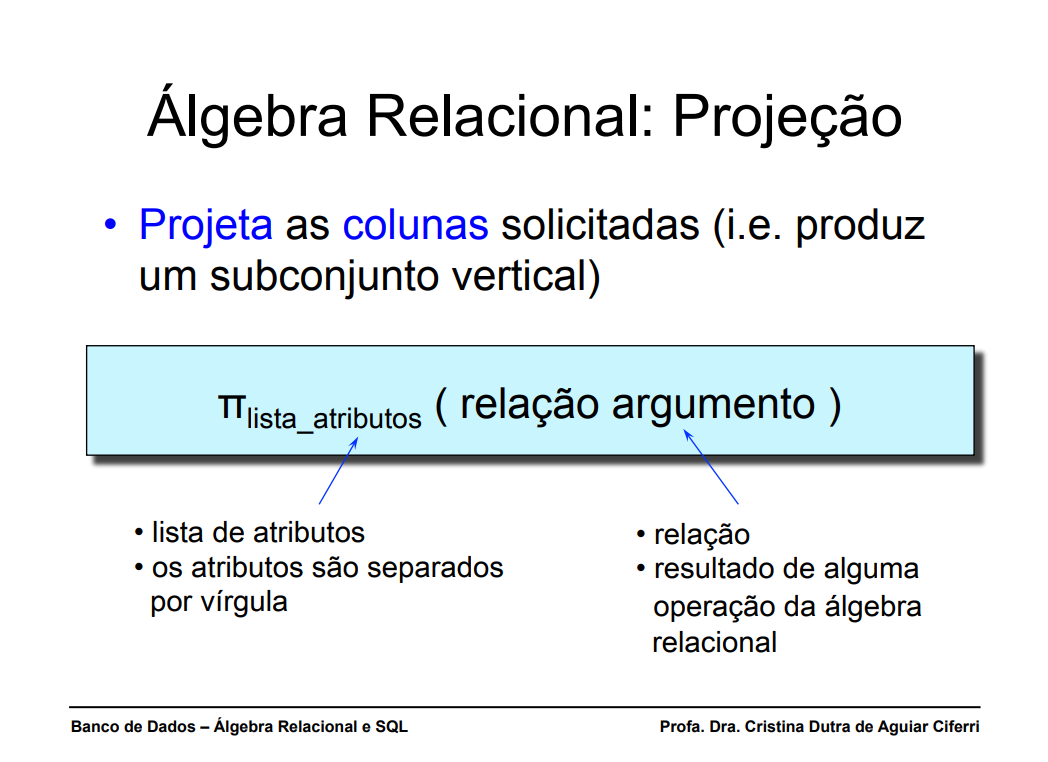
Como podemos ver na explicação acima a seleção pode no seu contexto envolver operadores de comparação e ter uma relação com outras seleções ou operações de algebra relacional.



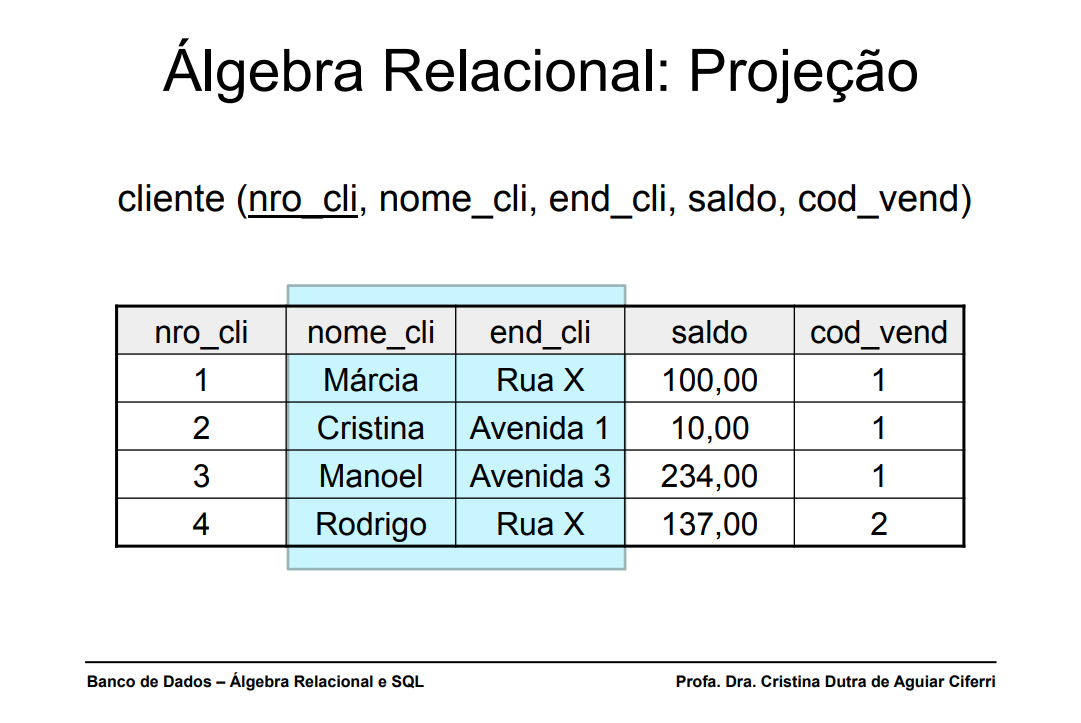
Já aqui temos um exemplo prático de uma tabela cliente que retorna 2 resultados de 4 possíveis a partir de um commando :  
**SELECT \* FROM CLIENTE WHERE nro\_cli = 2 or nro\_cli = 3**

**2.Projeção- π**

A operação de Projeção é recorrente nas mais diversas consultas sejam simples ou complexas, feita por profissionais ou amadores, a diferença para a seleção é que na projeção selecionamos as colunas de nosso interesse a serem consultadas.

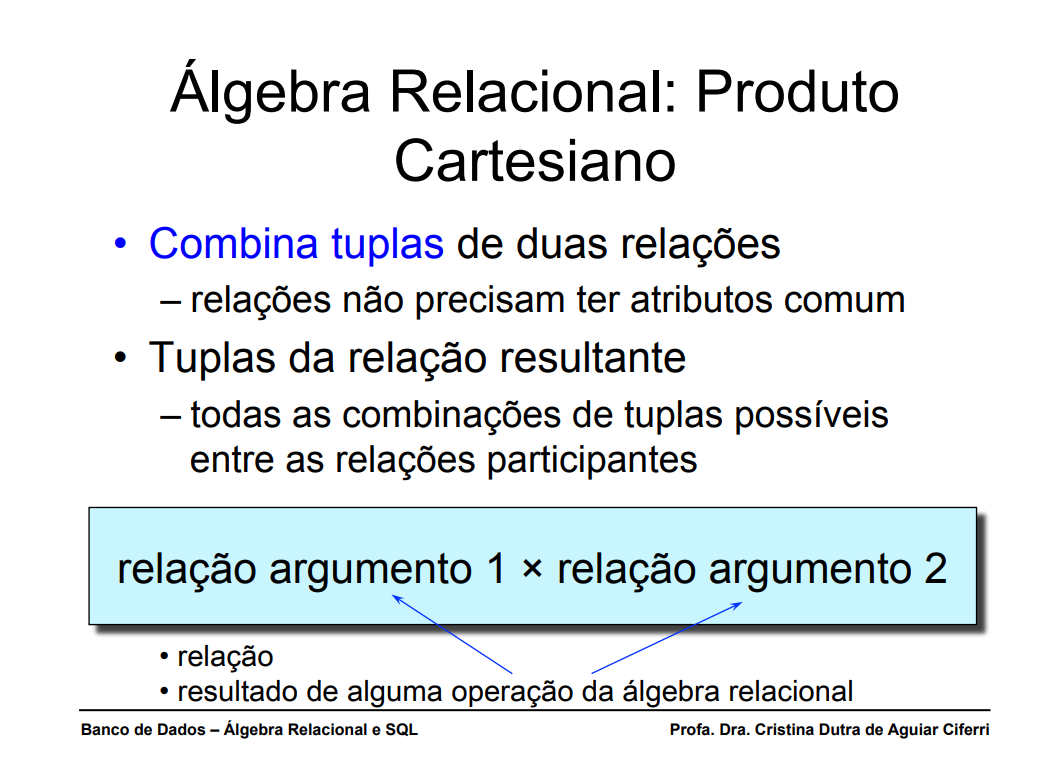


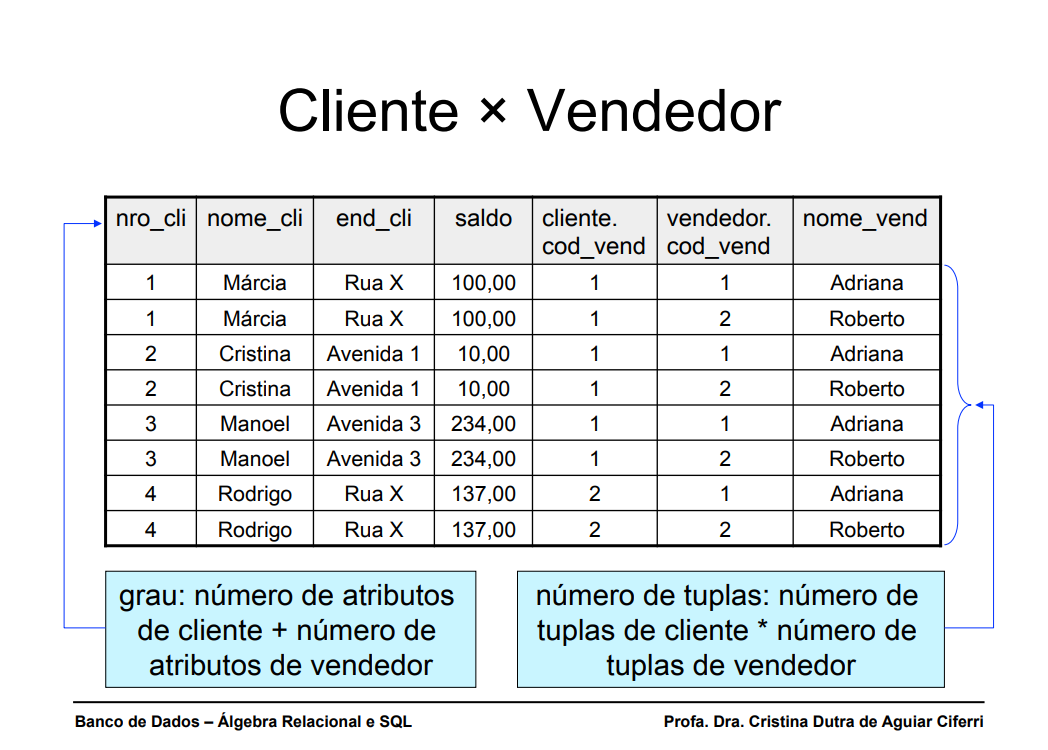
Na imagem acima temos um exemplo teórico da representação da projeção, com as colunas solicitadas representadas pela lista de atributos, isso produz justamente um subconjunto vertical.

****

Dentre as 5 opções de coluna da tabela cliente selecionamos 2 colunas que são elas nome\_cli e end\_cli representada no comando:  
**SELECT nome\_cli, end\_cli from CLIENTE**

**3.Produto Cartesiano - ×**

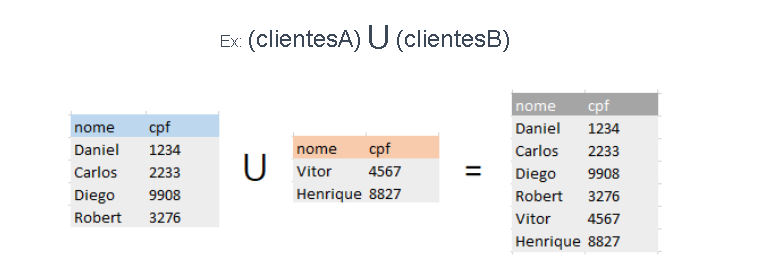
O produto cartesiano também é bem utilizado em consultas, ele consiste na relação de A x B que formam um produto C. 

Essa combinação de tuplas gera outra resultante como vemos no exemplo:

As tuplas de vendedor se propagam apesar de estarem em menor quantidade em relação as de vendedor.

**4.União - ⋃**

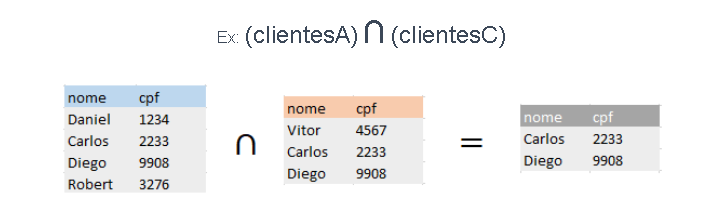
A união de AUB, traz um resultante que é C, porém diferente do plano cartesiano não há repetição de tuplas o que melhora o desempenho em relação ao exemplo anterior.



A vantagem de desempenho é nítida em relação ao plano cartesiano que repete tuplas sequencialmente.

**5.Interseção - ∩**

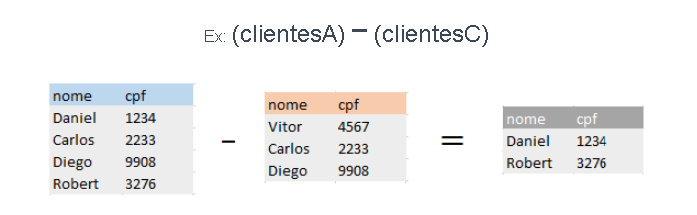
A interseção é a relação de A∩B com uma resultante C, porém aqui só temos as linhas em comum que existem nas duas tabelas.



Podemos notar que apenas Carlos e Diego com seus cpfs respectivos que fazem parte de ambas as tabelas foram exibidos como resultado da consulta

**6.Diferença de conjuntos -** −

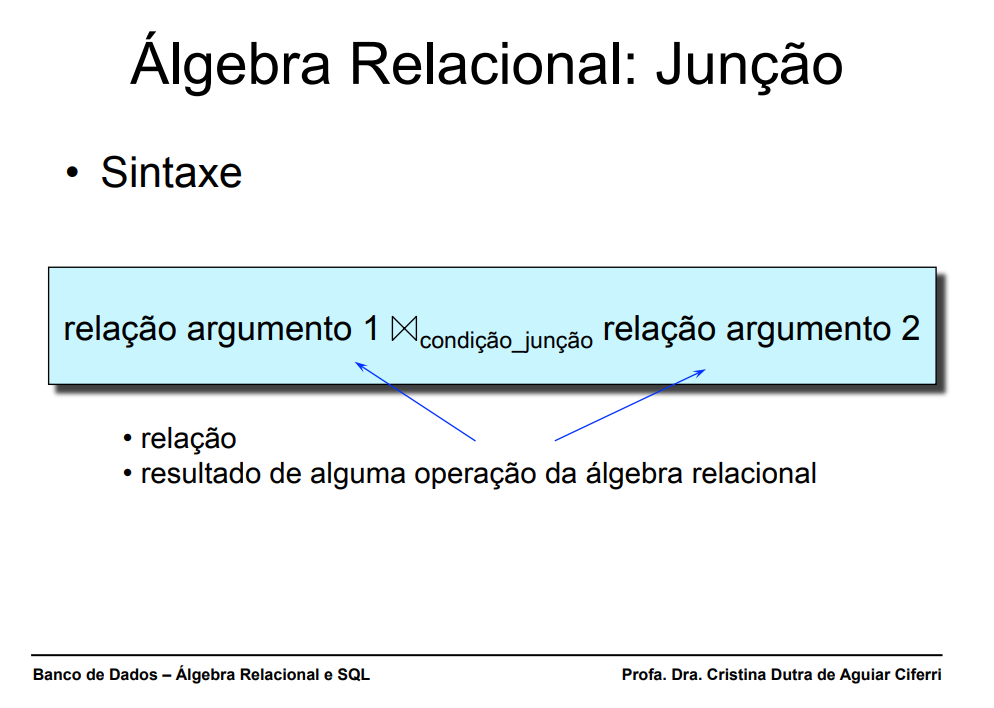
Como o nome já sugere aqui temos a diferença entre a relação de A-B com uma resultante C que são as tuplas que existem em A mas não em B.



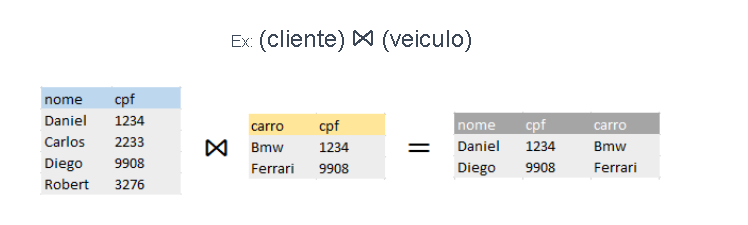
Daniel e Robert são nomes que não existem na Tabela B e por isso são os resultantes da tabela C nesse exemplo acima.

**7. Junção-** ⋈

A junção é um aprimoramento do plano cartesiano porque além de fazer uma combinação de tuplas ela também seleciona e por fim faz uma projeção, para remoção de duplicatas.



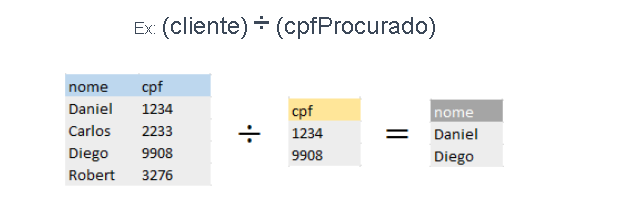
É uma das operações mais usadas no SQL por ter resultados precisos e bem definidos a partir das condições selecionadas.



Nesse exemplo temos um relaciomento entre a tabela A ⋈ B e a condição para que seja feita é que o cpf seja igual nas duas colunas exibindo tanto informação de nome que vem da tabela A e de carro que vem da Tabela B , tendo como resultado a tabela C.

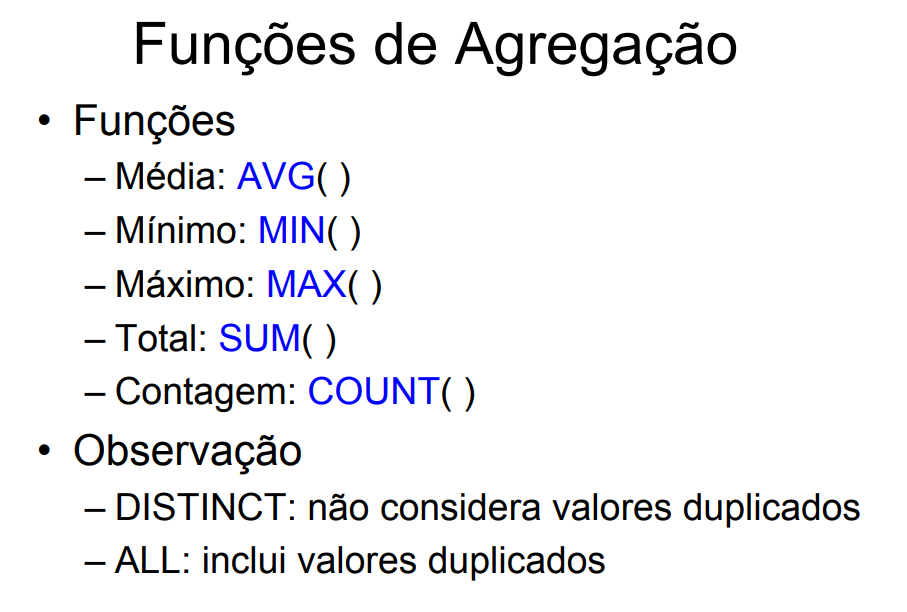
**8. Divisão- ÷**

Essa operação é bem simples já que a relação da divisão entre A÷B gera a Tabela C que exibe os campos comuns entre A e B como vemos no exemplo a seguir:

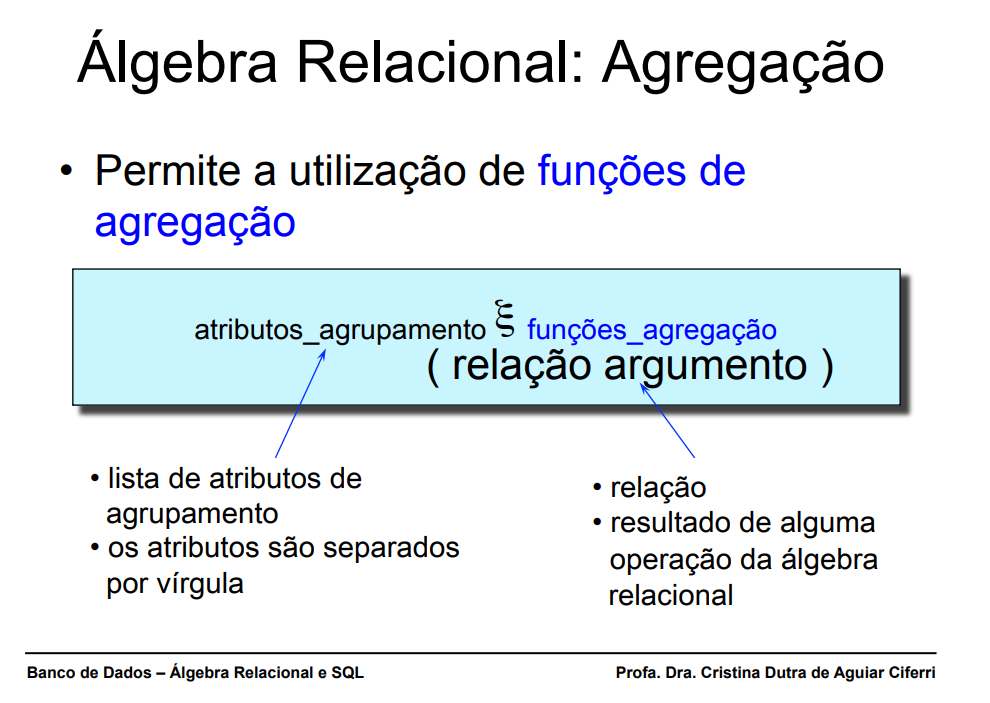
****

**9. Agregação- Γ**

As funções de agregação são extremamente utilizadas no dia a dia de um analista de dados elas permitem simplificar as relações de consultas. Dentre elas temos:



Dentre suas principais características temos que elas recebem uma coleção de valores e retornam um único valor como resultado, sendo numéricos ou não numéricos.



A sua lógica é baseada em agrupamento uma lista de atributos é submetida a uma operação que gera um único resultado esperado.

**REFERÊNCIAS :**

[**https://www.ic.unicamp.br/~geovane/mo410-091/Ch04a-AlgR\_pt.pdf**](https://www.ic.unicamp.br/~geovane/mo410-091/Ch04a-AlgR_pt.pdf)

[**http://wiki.icmc.usp.br/images/2/2c/SCC578920131-algebraSQL.pdf**](http://wiki.icmc.usp.br/images/2/2c/SCC578920131-algebraSQL.pdf)[**https://spaceprogrammer.com/bd/aprendendo-as-principais-operacoes-da-algebra-relacional/**](https://spaceprogrammer.com/bd/aprendendo-as-principais-operacoes-da-algebra-relacional/)

[**https://pt.wikipedia.org/wiki/%C3%81lgebra\_relacional**](https://pt.wikipedia.org/wiki/%C3%81lgebra_relacional)

[**https://eadsalvador.ifba.edu.br/moodle/pluginfile.php/110876/mod\_resource/content/1/AlgebraRelacional.pdf**](https://eadsalvador.ifba.edu.br/moodle/pluginfile.php/110876/mod_resource/content/1/AlgebraRelacional.pdf)