

# ÁLGEBRA RELACIONAL – PARTE 2

---

Sergio Mergen

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Operações Derivadas

- São operações que não adicionam poder de expressividade à linguagem, mas que simplificam consultas frequentes:
  - Intersecção ( $\cap$ )
  - Junção Theta ( $\bowtie$ )
  - Divisão ( $\div$ )
  - Atribuição ( $\leftarrow$ )

# Sumário

- Operações derivadas
  - **Interseção**
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Intersecção

- Operação binária:  $\cap$
- São mantidas as tuplas comuns às duas relações de entrada
- As relações de entrada devem ser *Union Compatible*
- Equivale a cláusula **INTERSECT** do SQL

# Intersecção

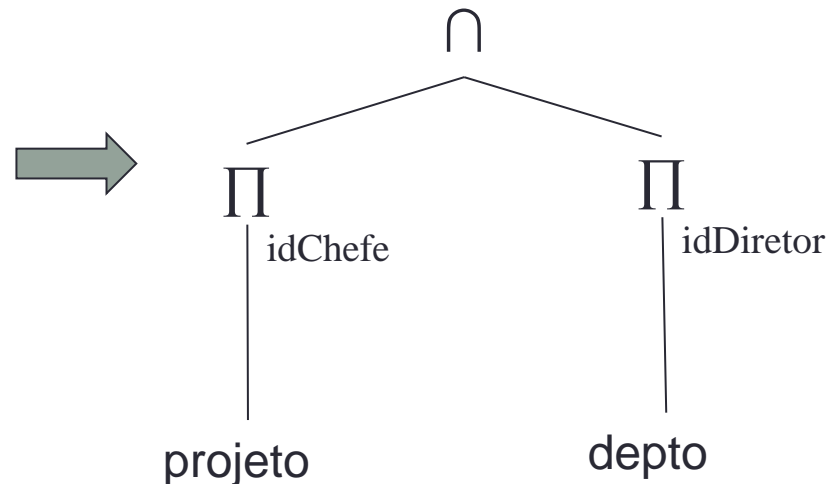
Listar os códigos de funcionários que são chefes e diretores

Funcionario		
idF	nome	idChefe
1	Marcos	null
2	Pedro	3
3	João	1

Depto		
idD	nome	idDiretor
1	TI	1
2	RH	2

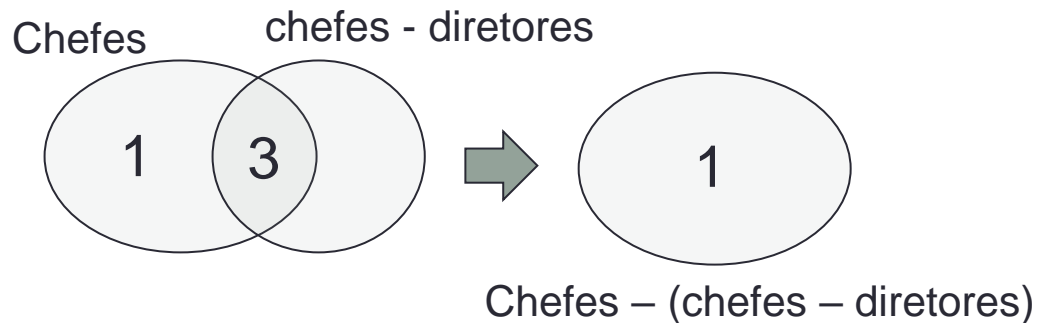
(select idChefe from funcionario)  
intersect  
(select idDiretor from depto)

Resposta	
idChefe	
1	



# Intersecção

Observação :  $r \cap s = r - (r - s)$



# Sumário

- Operações derivadas
  - Interseção
  - **Junção Theta**
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção



# Junção Theta

- Operação binária:  $\bowtie$
- A relação resultante é o produto cartesiano das duas relações de entrada
  - Seguida de uma seleção com base em critérios de junção theta
  - Ex.
$$\text{expressão1} \bowtie \text{expressão2}$$
$$\Theta \text{atr1 op atr2}$$
- Também chamada de Equi-Join
  - Quando a operação op é por igualdade
- Equivale a cláusula **JOIN ON** do SQL

# Junção Theta

Encontre o nome de cada departamento e o nome de todos os respectivos projetos desses departamentos

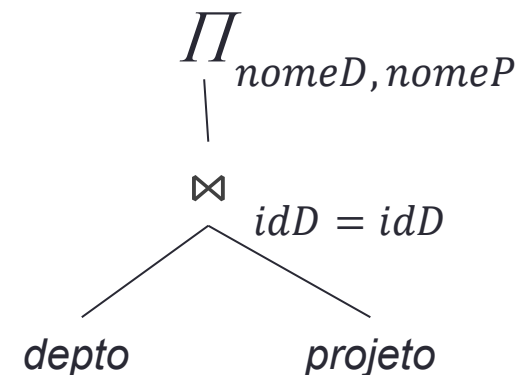
Projeto				
idP	nomeP	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

Deppto		
idD	nomeD	predio
1	TI	3
2	Marketing	2

```
select  nomeD, nomeP
from    depto d join projeto p on p. idD = d.idD
```

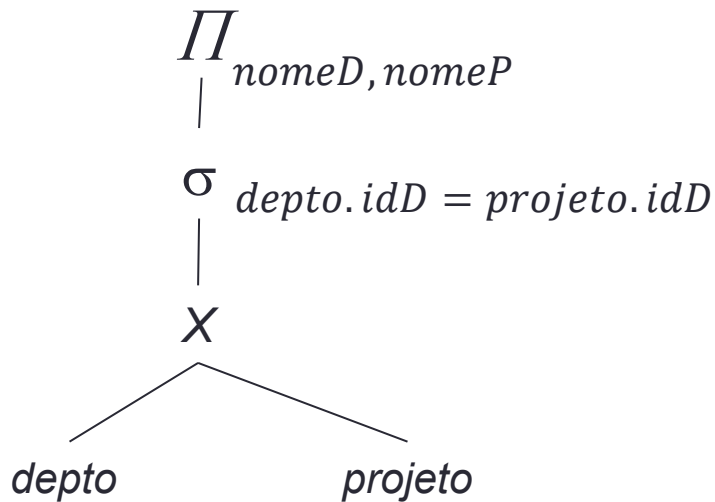


Resposta	
nomeD	nomeP
TI	ABC
TI	Lucrei

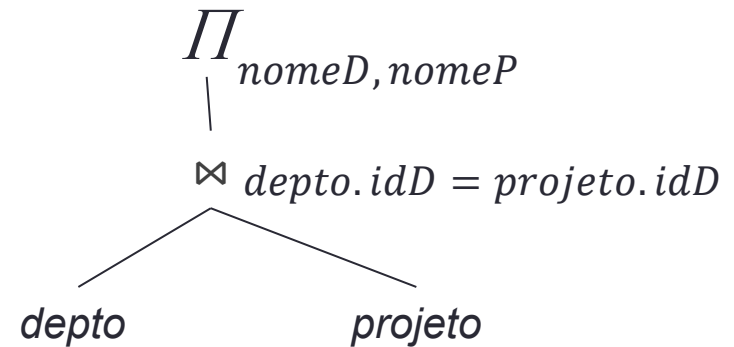


# Junção Theta

Permite encontrar o mesmo resultado que um produto cartesiano seguido de uma seleção



Usando produto cartesiano



Usando junção theta

# Junção Natural

- Operação binária:  $\bowtie$
- É uma junção theta onde não se especificam os critérios de junção
  - Assume-se que seja uma equi-join
  - Com base nos atributos de mesmo nome nas duas relações de entrada
- Equivale a cláusula **NATURAL JOIN** do SQL

# Junção Natural

Encontre o nome de cada departamento e o nome de todos os respectivos projetos desses departamentos

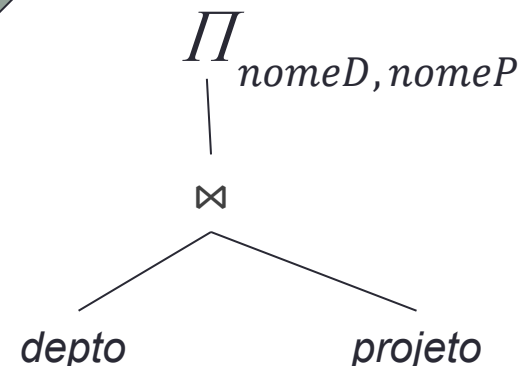
Projeto				
idP	nomeP	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

Deppto		
idD	nomeD	predio
1	TI	3
2	Marketing	2

```
select  nomeD, nomeP
from    depto natural join projeto
```

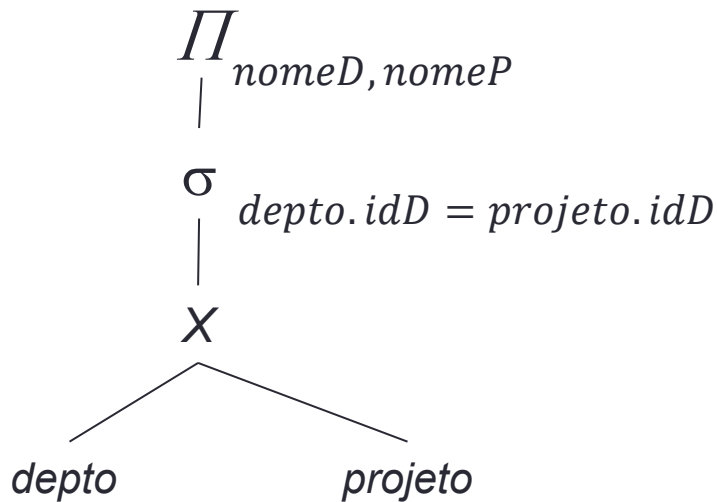


Resposta	
nomeD	nomeP
TI	ABC
TI	Lucrei

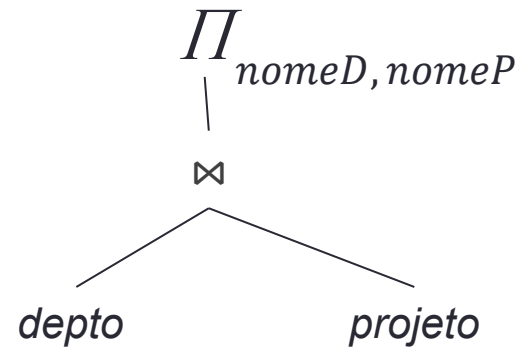


# Junção Natural

Permite encontrar o mesmo resultado que um produto cartesiano seguido de uma seleção



Usando produto cartesiano



Usando junção natural

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Divisão

- Operação binária:  $\div$
- A segunda relação (da direita) deve ser um subconjunto da primeira relação
  - Quanto aos seus atributos
- Os atributos sobrepostos devem ser compatíveis
- Não possui equivalente em SQL



# Divisão

- Mantem as tuplas da primeira relação que tiverem relação com todas as tuplas da segunda relação
  - A relação é determinada pelos atributos sobrepostos das duas relações
  - São retornados os atributos da primeira relação que não existirem na segunda relação
- Indicado para consultas do tipo “Para todo”

# Divisão

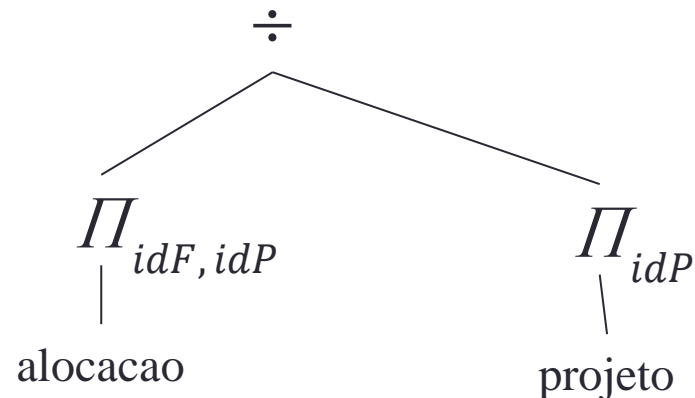
Encontre os ids dos funcionários que atuaram **em todos** os projetos

Funcionário	
idF	nome
1	Zé
2	João
3	Ana

Projeto	
idP	Nome
1	ABC
2	Caos

Alocação		
idP	idF	Função
1	1	Analista
1	2	Líder
2	3	Analista
2	2	Líder

Resposta
idFunc
2



# Divisão

- Pode ser obtida através das operações primitivas
  - $r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$
  - Onde:
    - $r$  e  $s$  são relações
    - $R$  são os atributos de  $r$
    - $S$  são os atributos de  $s$

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - **Atribuição**
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Atribuição

- Operação unária: ←
- Atribui a relação resultante de uma expressão para uma variável temporária
- Permite simplificar expressões complexas, através de refatorações
- Em SQL, o mesmo efeito pode ser obtida através de uma linguagem de programação
  - Ex. **PL-SQL**

# Atribuição

Exemplo: Descobrir funcionários que já trabalharam como líderes de projetos e que recebam mais do que 10.000 reais

**Expressão complexa:**

$\Pi_{\text{Nome}} (\sigma_{funcao = 'Lider'} (alocacao \bowtie (\sigma_{salario > 10.000}(\text{funcionario}))))$

**Expressão refatorada:**

$\text{BemPagos} \leftarrow \sigma_{salario > 10.000}(\text{funcionario})$

$\text{AlocacaoBemPagos} \leftarrow alocacao \bowtie \text{BemPagos}$

$\text{AlocacaoLideresBemPagos} \leftarrow \sigma_{funcao = 'Lider'}(\text{AlocacaoBemPagos})$

$\Pi_{\text{Nome}}(\text{AlocacaoLideresBemPagos})$

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Operações adicionadas

- As operações abaixo foram criadas para aumentar o poder de expressão da álgebra relacional
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção



# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - **Projeção generalizada**
  - Funções de agregação
  - Outer Join
  - Semi Join
  - Anti join

# Projeção Generalizada

- Estende a operação de projeção permitindo que funções aritméticas sejam usadas na lista de projeção.
- Notação:  $\Pi_{F_1, F_2, \dots, F_n}(E)$
- *Onde:*
  - $E$  é uma expressão
  - $F_i$  são funções sobre  $E$ , envolvendo atributos, constantes e operações matemáticas
- Em SQL, pode-se fazer o mesmo

# Projeção Generalizada

Exemplo: encontre os nomes e a duração de todos projetos, em meses

Projeto				
id	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

**select** *nome, duracao \* 12*  
**from** *projeto*



$\Pi$   
nome, duracao \* 12  
projeto

Resposta	
nome	Duracao * 12
ABC	36
Lucrei	24

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - **Agregação**
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Agregação

- Operação unária: G
- Aplica uma função de agregação sobre tuplas da relação de entrada.
  - Avg: média dos valores
  - Min: valor mínimo
  - Max: valor máximo
  - Sum: soma dos valores
  - Count: número de valores
- Equivale a cláusula **GROUP BY** do SQL

# Agregação

- **Notação**

$$g_1, g_2, \dots, g_k \overset{G}{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$$

- *Onde:*

- $E$  é uma expressão
  - $g_1, g_2, \dots, g_k$  é a lista dos atributos a agrupar (pode ser vazia)
  - Cada  $F_i$  é uma função de agregação
  - Cada  $A_i$  é um nome de atributo
- 
- Convém renomear o atributo que guarda o resultado de uma função de agregação
    - Para simplificar, pode-se usar o **AS**

# Agregação

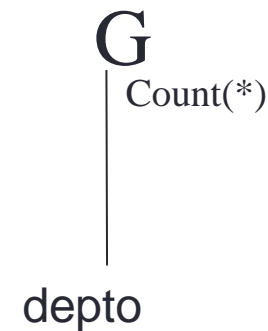
Calcular o número de departamentos.

Deppto		
idD	nome	idDiretor
1	TI	1
2	RH	5
3	Marketing	3

```
select count (*)  
from depto
```



Resposta
Count(*)
3



# Agregação

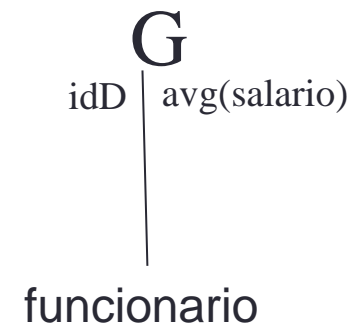
Ex. Listar o salário médio por departamento.

Funcionario			
idF	nome	Salario	idD
1	Marcos	10000	1
2	Ana	6000	2
3	João	5000	1

**select**     *idD, avg (salario)*  
**from**        *funcionario*  
**group by** *idD*



Resposta	
idD	avg (salario)
1	7500
2	6000





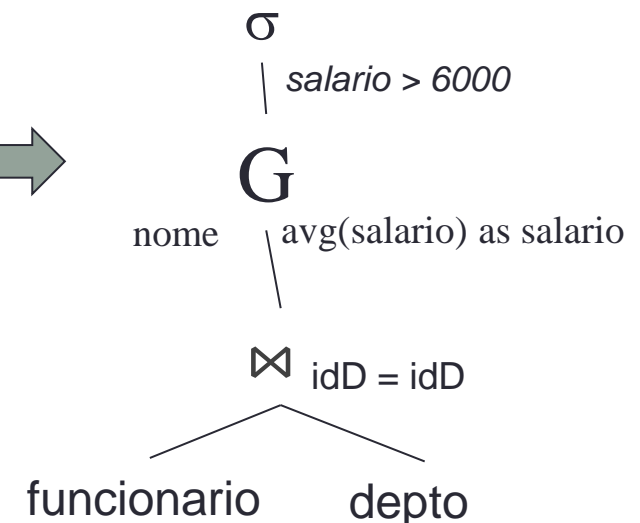
# Agregação

Ex. Listar os nomes de todas os departamentos cuja média de salário seja superior a R\$6.000.

Funcionario			
idF	nomeF	salario	idD
1	Marcos	10000	1
2	Ana	6000	2
3	João	5000	1

Depto			
idD	nome	predio	idDir
1	TI	3 - centro	1
2	RH	2 - norte	2

**select**      *nome, avg (salario)*  
**from**        *funcionario f join depto d using (idD)*  
**group by**    *nome*  
**having**       *avg (salario) > 6000*



Resposta	
nome	Avg(salario)
TI	7500

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - **Junção Externa**
  - Semi-junção
  - Anti-junção

# Junção Externa

- Considerando que uma junção tem duas partes
  - Parte principal
  - Parte secundária
- A junção externa traz todos os registros da parte principal
  - Mesmo que alguns desses registros não tenham correspondência com nenhum registro da parte secundária
  - Registros sem correspondência são complementados com valores NULOS

# Junção Externa

- Estratégias de junção externa
  - Junção Externa à esquerda ( $\bowtie$ )
    - A parte principal fica do lado externo da junção
    - Ou seja, todos registros do lado externo são retornados
  - Junção Externa à direita ( $\bowtie$ )
    - A parte principal fica do lado interno da junção
    - Ou seja, todos registros do lado interno são retornados
  - Junção Externa Completa ( $\bowtie$ )
    - todos registros dos dois lados são retornados
- Na ausência de critérios de junção
  - São consideradas as colunas de mesmo nome
    - Como na Junção Natural

# Junção Externa

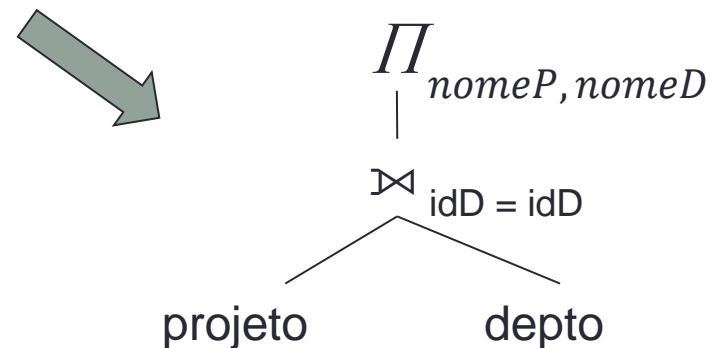
Ex. Encontre todos projetos e seus departamentos, se houver

Projeto				
idP	nomeP	duracao	custo	idD
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	2

Deppto		
idD	nomeD	predio
2	Marketing	2
3	RH	2

**select**     *nomeP, nomeD*  
**from**       *projeto p left join depto d on p.idD = d.idD*

Resposta	
nomeP	nomeD
Lucrei	Marketing
Caos	Marketing
Genesis	null



# Junção Externa

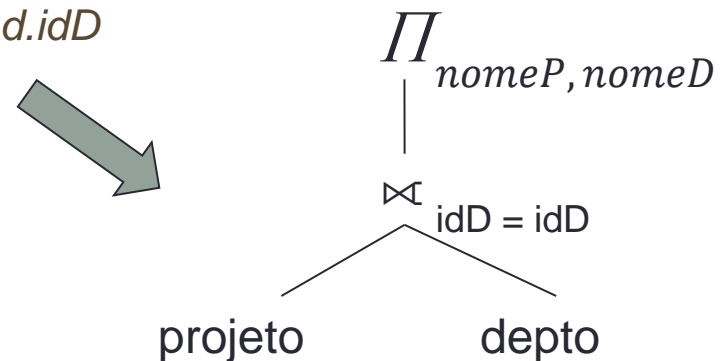
Ex. Encontre todos departamentos e seus projetos, se houver

Projeto				
idP	nomeP	duracao	custo	idD
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	2

Depto		
idD	nomeD	predio
2	Marketing	2
3	RH	2

**select**     *nomeP, nomeD*  
**from**       *projeto p right join depto d on p.idD = d.idD*

Resposta	
nomeP	nomeD
Lucrei	Marketing
Caos	Marketing
null	RH



# Junção Externa

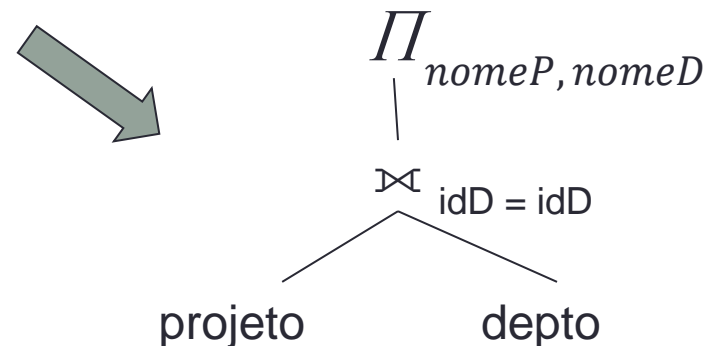
Ex. Encontre todos departamentos e projetos, e suas associações

Projeto				
idP	nomeP	duracao	custo	idD
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	2

Depto		
idD	nomeD	predio
2	Marketing	2
3	RH	2

**select**     *nomeP, nomeD*  
**from**       **projeto full outer join** *depto on p.idD = d.idD*

Resposta	
nomeP	nomeD
Lucrei	Marketing
Caos	Marketing
Genesis	null
null	RH



# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - **Semi-junção**
  - Anti-junção



# Semi-junção

- Considerando que uma junção tem duas partes
  - parte principal
  - parte secundária
- Em uma semi-junção
  - Apenas tuplas da parte principal são retornadas
  - Cada tupla pode ser retornada apenas uma vez
    - Caso haja correspondências com a parte secundária

# Semi-junção

- As estratégias de semi-junção podem ser classificadas em
  - Semi-junção à esquerda ( $\bowtie$ )
    - A parte principal fica do lado externo (esquerdo) da junção
  - Semi-junção à direita ( $\bowtie$ )
    - A parte principal fica do lado interno (direito) da junção

# Semi-junção

- Em SQL, sem-junções são expressas na forma de subconsultas
  - A parte principal é a consulta principal
  - A parte secundária é a subconsulta
- Os exemplos abaixo mostram duas formas de uso (EXISTS e IN)
  - Parte principal: projeto
  - Parte secundária: depto

```
SELECT nome  
FROM projeto p WHERE EXISTS  
    (SELECT 1 FROM depto d  
     WHERE p.idD = d.idD )
```

```
SELECT nome  
FROM projeto WHERE idD IN  
    (SELECT idD FROM depto)
```

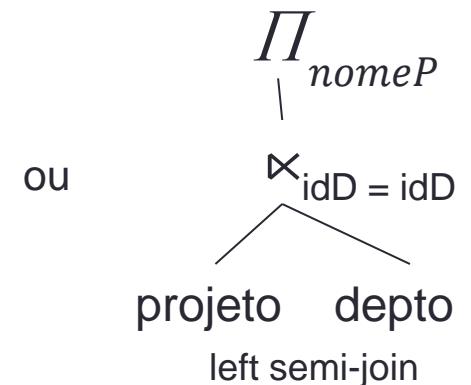
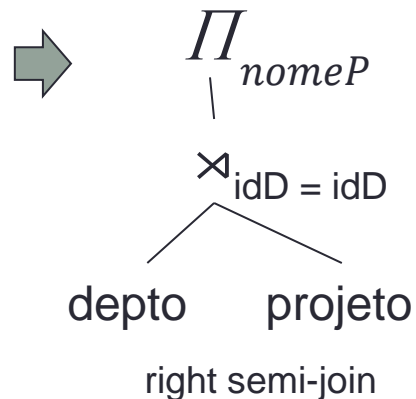
# Semi-junção

Ex. Encontre todos projetos que possuam departamento

Projeto				
idP	nomeP	duracao	custo	idD
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	2

Deppto		
idD	nomeD	predio
2	Marketing	2
3	RH	2

```
select  nomeP
from    projeto where idD in (
    select idD
    from  Deppto
)
```



Resposta
nomeP
Lucrei

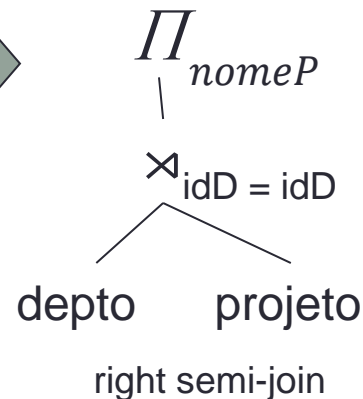
# Semi-junção

Ex. Encontre todos projetos que possuam departamento

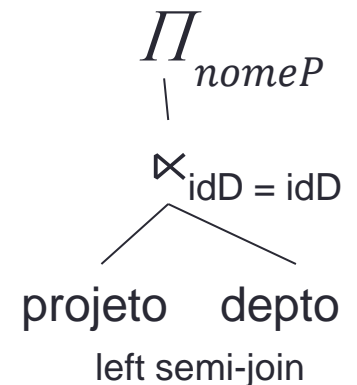
Projeto				
idP	nomeP	duracao	custo	idD
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	null
4	Caos	10	100.000	2

Deppto		
idD	nomeD	predio
2	Marketing	2
3	RH	2

```
select    nomeP
from      projeto p where exists (
  select 1 from depto d
  where d.idD = p.idD
)
```



ou



Resposta
nomeP
Lucrei

# Sumário

- Operações derivadas
  - Interseção
  - Junção Theta
  - Divisão
  - Atribuição
- Operações adicionadas
  - Projeção generalizada
  - Funções de agregação
  - Junção Externa
  - Semi-junção
  - Anti-junção

# Anti-junção

- Considerando que uma junção tem duas partes
  - parte principal
  - parte secundária
- Em uma anti-junção
  - Apenas tuplas da parte principal são retornadas
  - Cada tupla pode ser retornada apenas uma vez
    - Caso não haja correspondências com a parte secundária

# Anti-junção

- As estratégias de anti-junção podem ser classificadas em
  - Anti-junção à esquerda ( $\triangleright$ )
    - Retorna tuplas do lado externo (esquerdo), desde que não possuam correspondência com nenhuma tupla do lado interno (direito)
  - Anti-junção à direita ( $\triangleleft$ )
    - Retorna tuplas do lado interno(direito), desde que não possuam correspondência com nenhuma tupla do lado externo (esquerdo)



# Anti-junção

- Em SQL
    - Anti-junções são expressas na forma de subconsultas com NOT EXISTS
      - A parte principal é a consulta principal
      - A parte secundária é a subconsulta
    - Exemplo
      - Parte principal: projeto
      - Parte secundária: depto
- ```
SELECT nome
FROM projeto p WHERE NOT EXISTS
    (SELECT 1 FROM depto d
     WHERE p.idD = d.idD )
```
- Obs. NOT IN não necessariamente é igual a NOT EXISTS

# Anti-junção

Ex. Encontre todos projetos que não possuam departamento

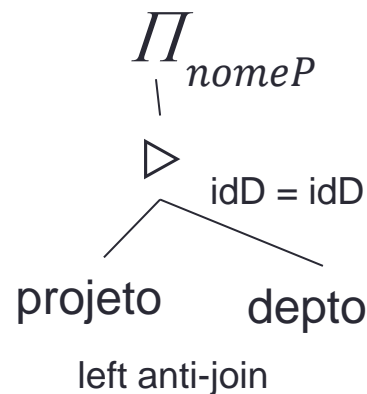
| Projeto |         |         |         |      |
|---------|---------|---------|---------|------|
| idP     | nomeP   | duracao | custo   | idD  |
| 2       | Lucrei  | 2       | 30.000  | 2    |
| 3       | Genesis | 2       | 15.000  | null |
| 4       | Caos    | 10      | 100.000 | 2    |

| Depto |           |        |
|-------|-----------|--------|
| idD   | nomeD     | predio |
| 2     | Marketing | 2      |
| 3     | RH        | 2      |

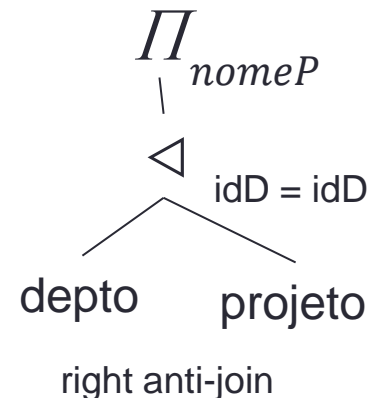
```
select  nomeP
from    projeto p where not exists (
  select 1 from  Depto d
  where d.idD = p.idD )
```



| Resposta |
|----------|
| nomeP    |
| Genesis  |
| Caos     |



ou



# Atividade Individual

Os exercícios a seguir são baseados em dados representados pelo esquema abaixo

movie (movie\_id, title)

person (person\_id, person\_name)

Movie\_cast (movie\_id, person\_id, character\_name, cast\_order)

movie\_id referencia movie

person\_id referencia person

# Atividade Individual

Com base no modelo relacional do slide anterior, descreva as consultas em SQL e em álgebra relacional que respondam os itens abaixo:

1. Número de filmes lançados após o ano 2000
2. Ano médio dos filmes do Bruce Willis
3. Mostrar a quantidade de filmes por ano, para quantidades maiores ou iguais a 5
4. Para os filmes de 2007, mostrar o título e os nomes dos personagens. Mostrar também os filmes desse ano que não tenham personagens definidos
5. Descobrir os nomes dos filmes que não tenham artistas definidos

# Atividade Individual

- Use a ferramenta DBest para fazer os exercícios
- Use apenas as operações vistas na aula de hoje
  - Elas estão presentes na aba “Rel. Algebra Operators”
- Use os .dat disponíveis no moodle para importar as tabelas
- Entregue as árvores em formato de consulta
  - A ferramenta possui opção para exportar nesse formato (Save Query)

# Atividade Individual

- Relação entre operadores do DDest e os da álgebra relacional vistos na aula de hoje

| Álgebra Relacional        | DBest                |
|---------------------------|----------------------|
| Agregação com agrupamento | Hash Group           |
| Agregação sem agrupamento | Aggregation          |
| Junção Theta              | Join                 |
| Junção Externa à esquerda | Left Outer Join      |
| Junção Externa à direita  | Right Outer Join     |
| Semi-junção à esquerda    | Semi Join            |
| Semi-junção à direita     | Hash Right Semi Join |
| Anti-junção à esquerda    | Anti Join            |
| Anti-junção à direita     | Hash Right Anti Join |
| Interseção                | Hash Intersection    |