

# Processamento de Consultas

## Banco de Dados: Teoria e Prática

André Santanchè e Patrícia Cavoto  
Instituto de Computação - UNICAMP  
Setembro 2016

# Pensador Profundo

- Qual a resposta para a vida, o universo e tudo mais?

(Guia do Mochileiro das Galáxias)

# Pensador Profundo

- Qual a resposta para a vida, o universo e tudo mais?
  - ... 7,5 milhões de anos depois  
(Guia do Mochileiro das Galáxias)

# Pensador Profundo

- Qual a resposta para a vida, o universo e tudo mais?
  - ... 7,5 milhões de anos depois
  - ... a resposta é:

(Guia do Mochileiro das Galáxias)

# Pensador Profundo

- Qual a resposta para a vida, o universo e tudo mais?
  - ... 7,5 milhões de anos depois
  - ... a resposta é: 42

(Guia do Mochileiro das Galáxias)

# Data Deluge

- Genoma Humano
  - 3.3 bilhões base-pairs
- Facebook
  - 30/06/2015 - 1,49 bilhões de usuários ativos
    - <http://newsroom.fb.com/company-info/>

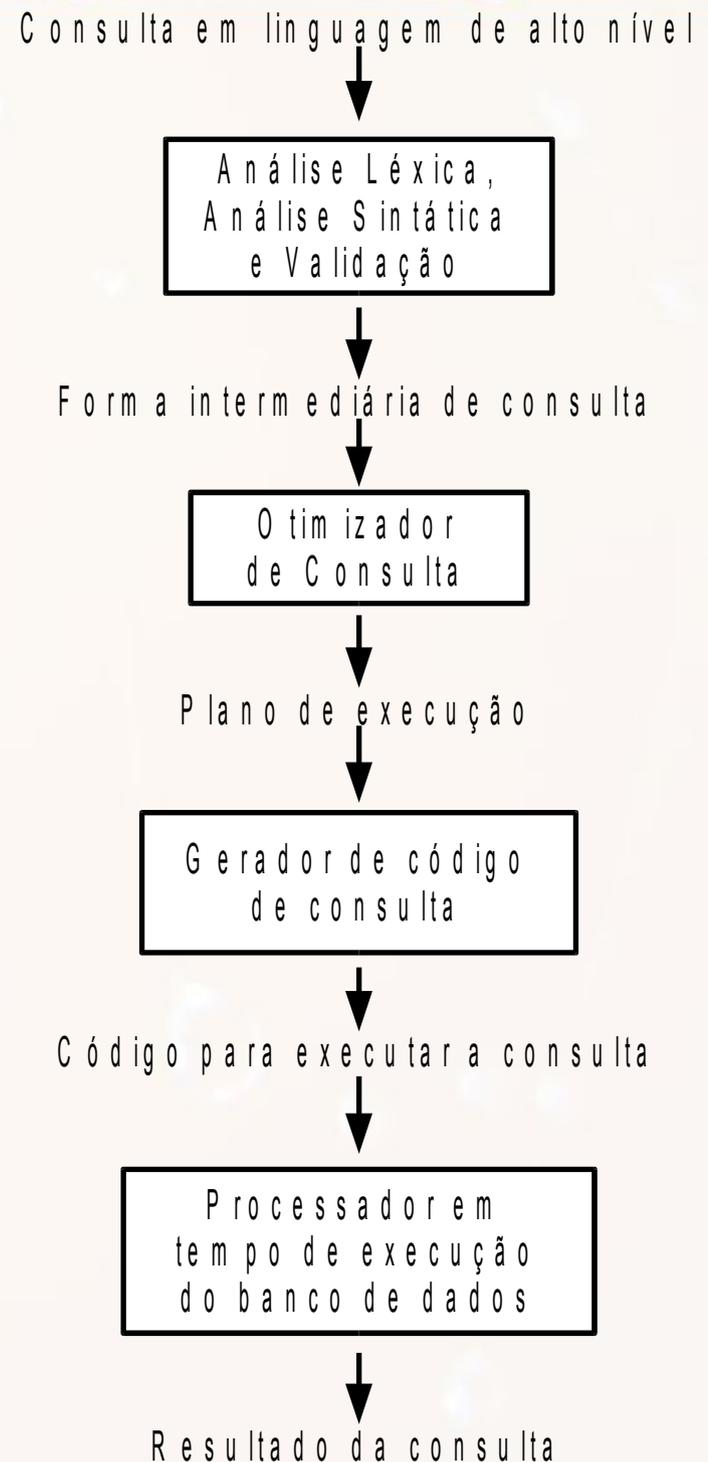
# Lei de Moore

- Poder de processamento dobra a cada dois anos
- Como crescem os dados?

# Execução de Consulta

-

## Passos Típicos (Elmasri, 2010)



# Execução de Consulta

-

## Passos Típicos

Consulta em linguagem de alto nível

Análise Léxica,  
Análise Sintática  
e Validação

Forma intermediária de consulta

Otimizador  
de Consulta

Plano de execução

Gerador de código  
de consulta

Código para executar a consulta

Processador em  
tempo de execução  
do banco de dados

Resultado da consulta

# Análise e Validação

## ■ Análise e Validação

- Análise léxica
- Análise sintática
- Validação

## ■ Representações internas:

- árvore de consulta
- grafo de consulta

Consulta em linguagem de alto nível

Análise Léxica,  
Análise Sintática  
e Validação

Forma intermediária de consulta

Otimizador  
de Consulta

Plano de execução

Gerador de código  
de consulta

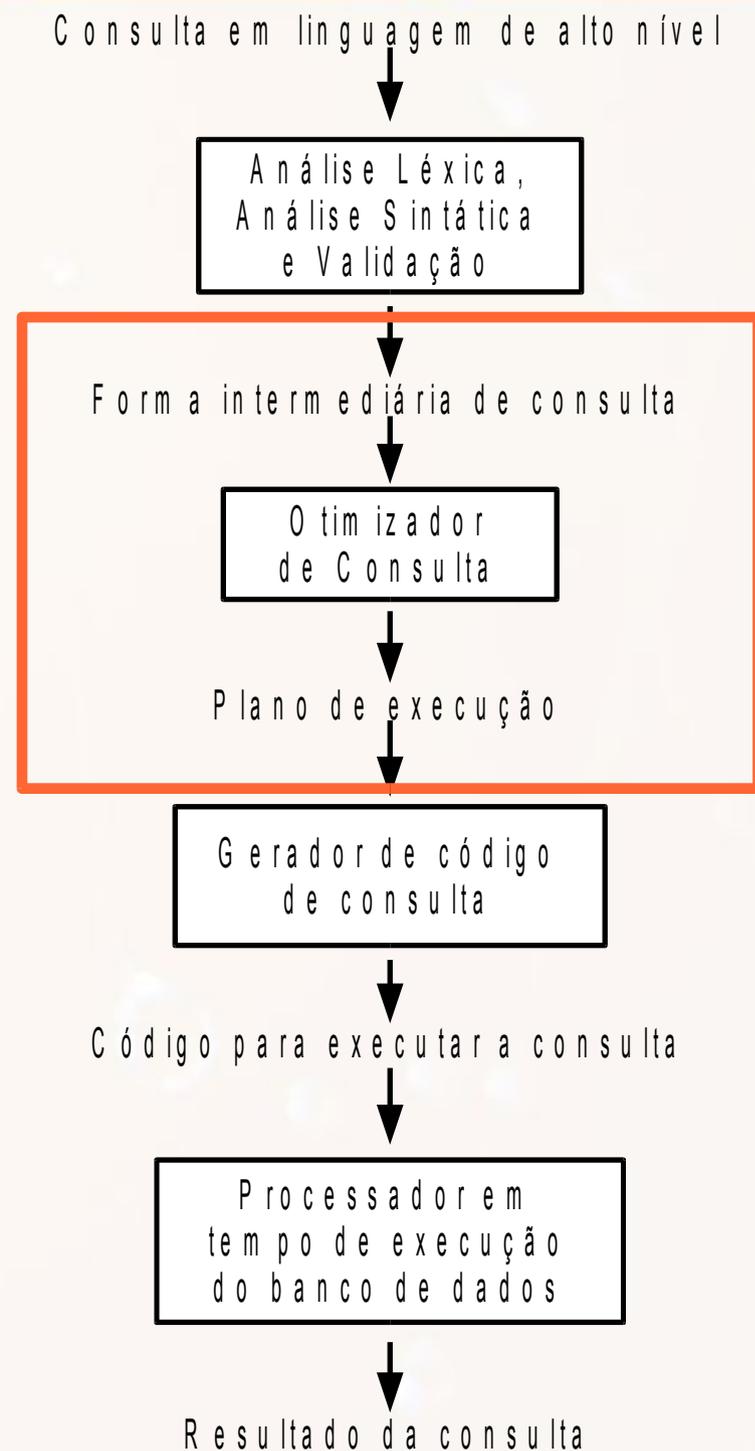
Código para executar a consulta

Processador em  
tempo de execução  
do banco de dados

Resultado da consulta

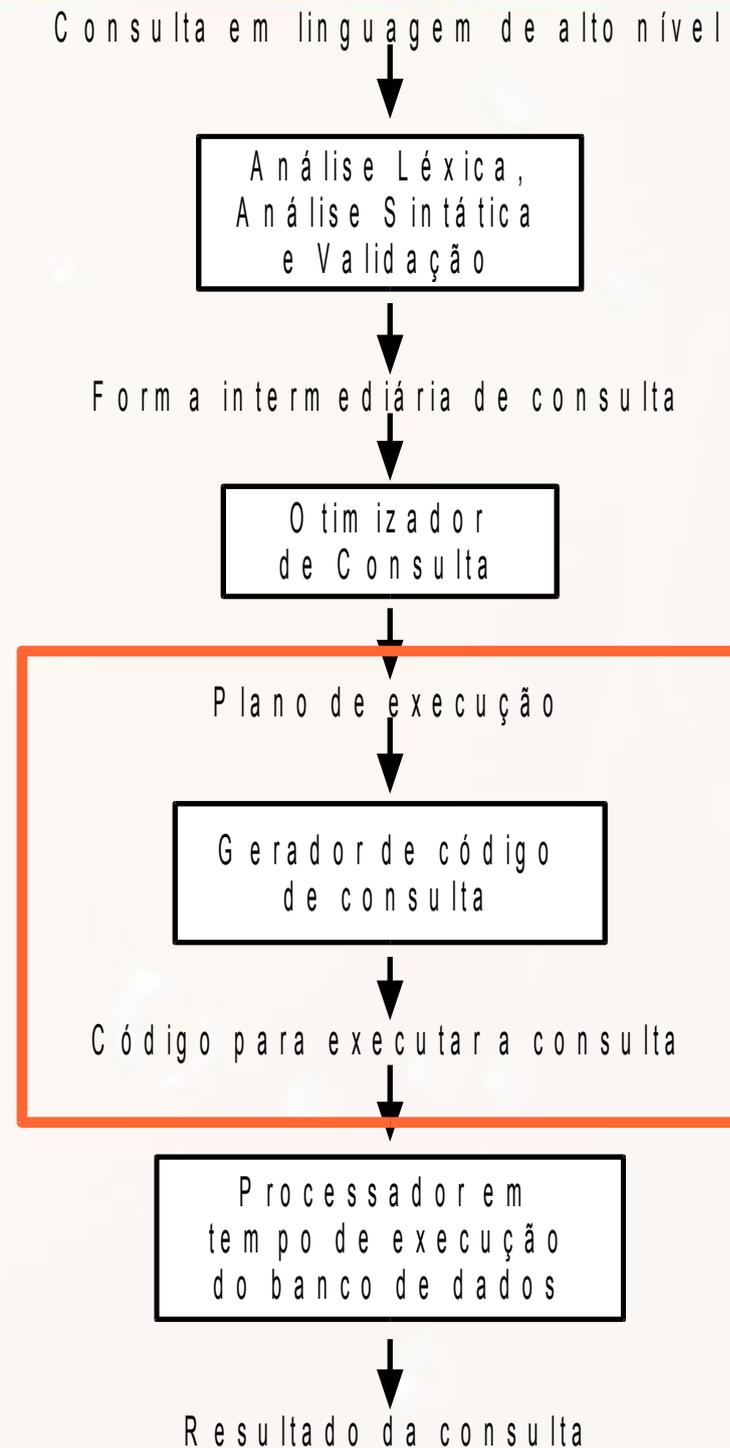
# Estratégia de Execução

- Consulta possui muitas estratégias de execução possíveis
- Planejamento da Estratégia de Execução
  - Otimização → processo de escolha da estratégia adequada (razoavelmente eficiente)



# Código da Consulta

- Pode ser:
  - Executado diretamente
    - modo interpretado
  - Armazenado e executado quando necessário
    - modo compilado



# Execução do Código

- Processador executa código da consulta
- Produz resultado da execução

Consulta em linguagem de alto nível

Análise Léxica,  
Análise Sintática  
e Validação

Forma intermediária de consulta

Otimizador  
de Consulta

Plano de execução

Gerador de código  
de consulta

Código para executar a consulta

Processador em  
tempo de execução  
do banco de dados

Resultado da consulta

**Ênfase desta aula:  
Otimização de Consultas**

# Consultas Declarativas

- “O quê” ao invés de “Como”
- Otimização de consulta
  - Solução razoavelmente eficiente (Elmasri, 2011)
  - Solução ótima pode ser muito custosa

# Consulta SQL em Álgebra Relacional

- Consulta SQL → Álgebra Relacional Estendida
  - Inclui operadores como COUNT, SUM e MAX
- Consulta SQL decomposta em blocos
  - Bloco de Consulta ou Bloco Simples:
    - Contém uma única expressão SELECT-FROM-WHERE (GROUP BY e HAVING se houver)
    - Sem aninhamento
  - Consultas aninhadas são identificadas como consultas independentes

# Decomposição em Blocos

## Exemplo

### ■ Tabela

Pessoa (Codigo, Nome, Telefone, AnoFiliacao)

### ■ Nome dos filiados mais antigos:

```
SELECT Codigo, Nome
FROM PESSOA
WHERE AnoFiliacao = (SELECT MIN(AnoFiliacao)
                     FROM PESSOA)
```

### ■ Blocos

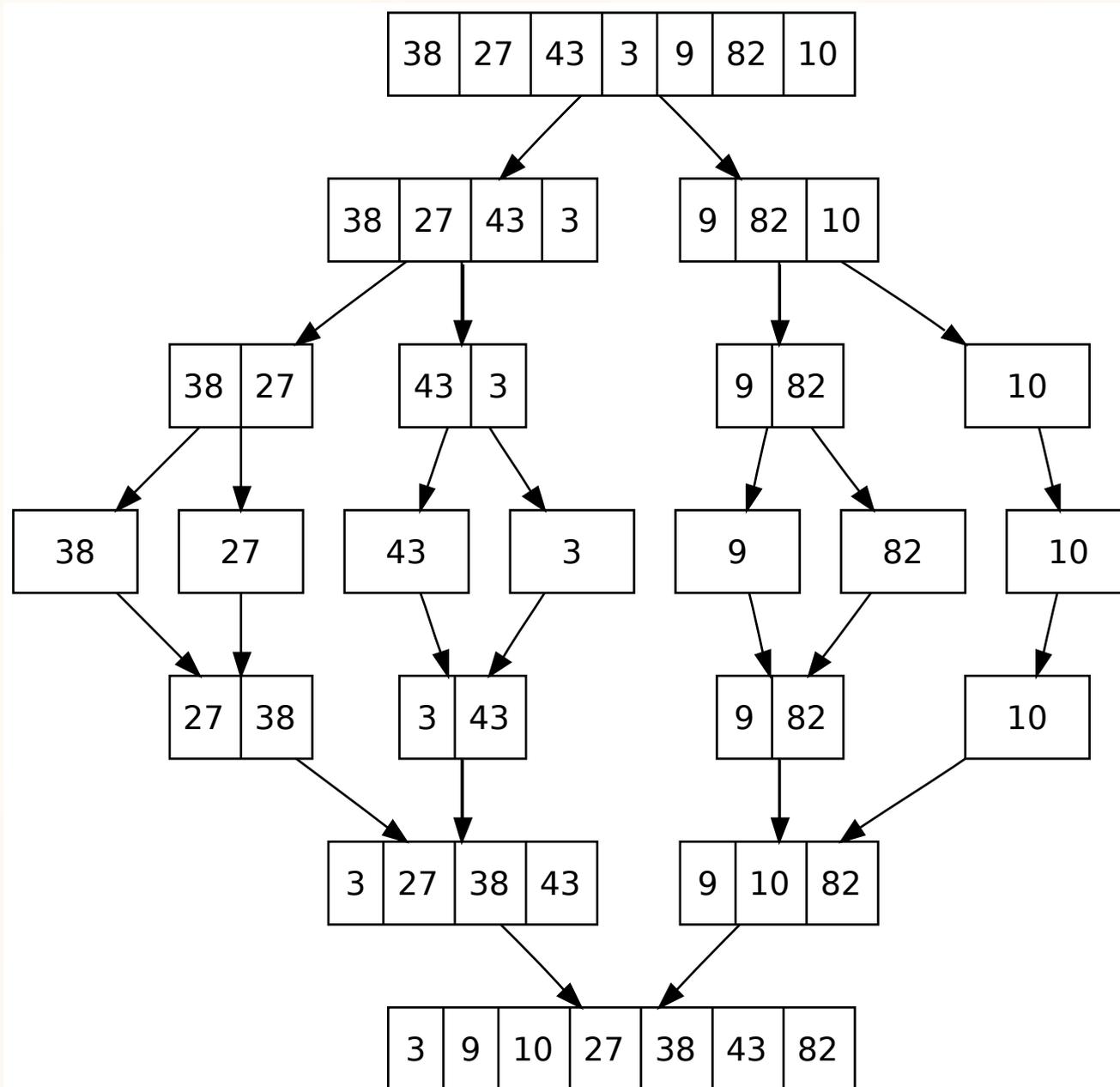
```
① SELECT Codigo, Nome
   FROM PESSOA
   WHERE AnoFiliacao = (referência ②)
```

```
② SELECT MIN(AnoFiliacao)
   FROM PESSOA
```

# Algoritmos para Operações

# Ordenação Externa

# Merge Sort Tradicional



# Ordenação Externa

5,9

7,2

8,4

1,6

3,6

9,1

5

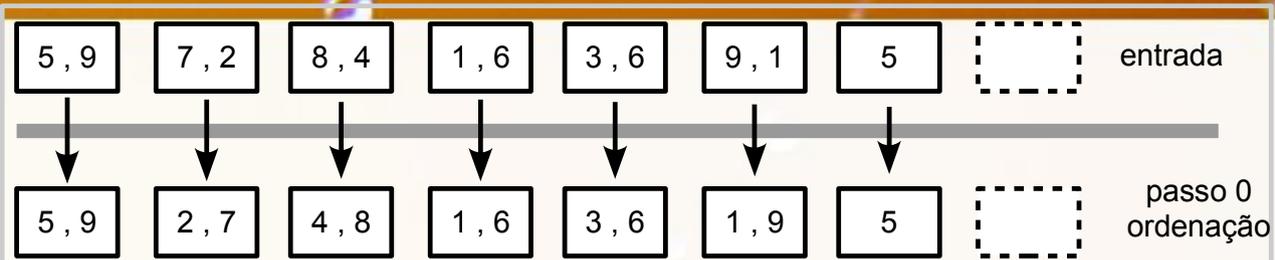


entrada

Entrada organizada em páginas de tamanhos iguais:

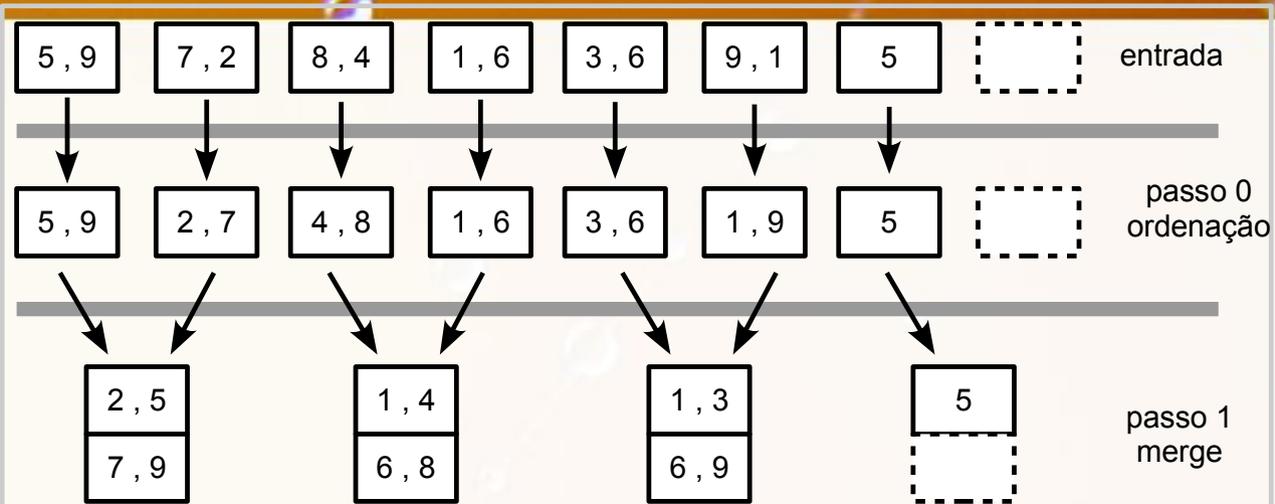
- 13 blocos de disco (bd)
- 3 blocos de memória (bm)

Exemplo  
Inspirado  
em  
(Ramakrishnan, 2013)



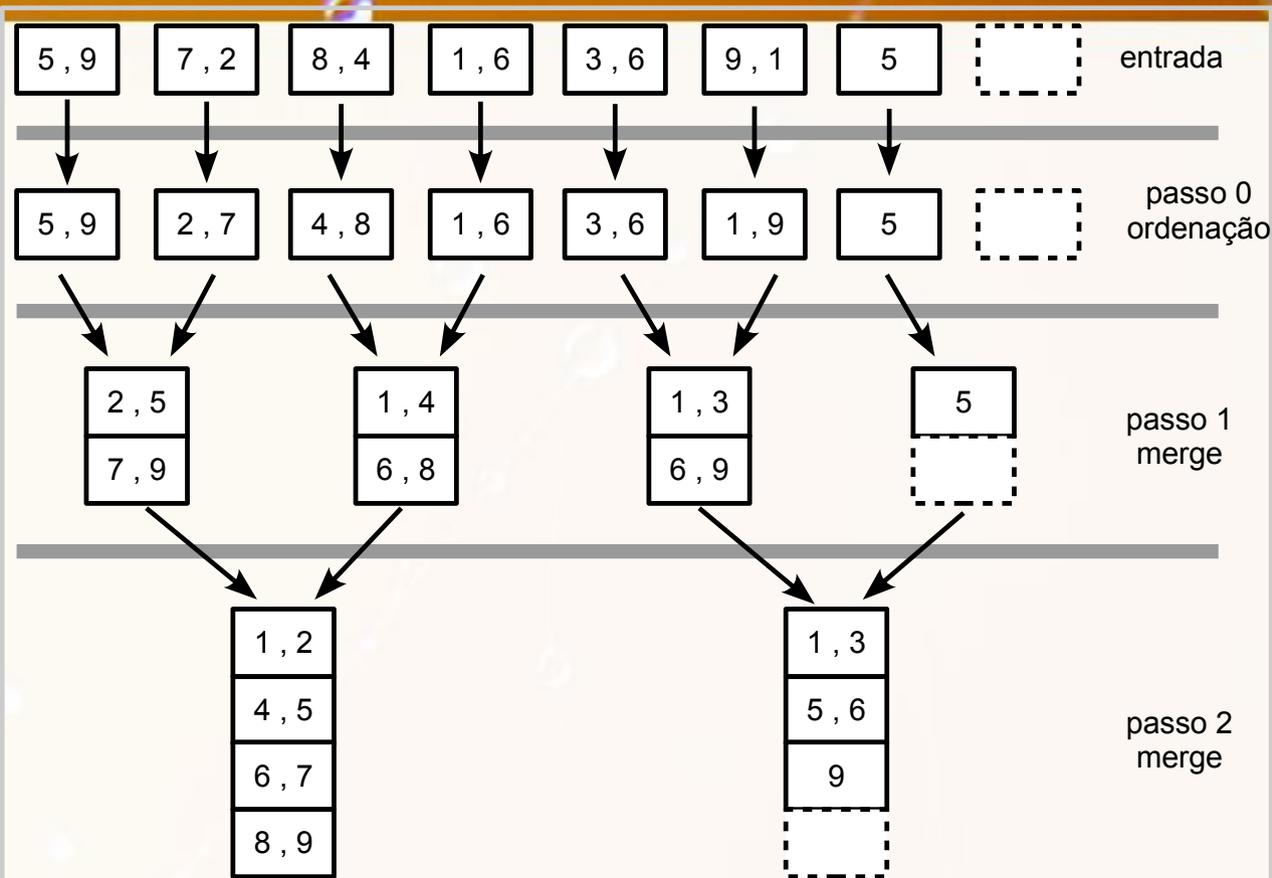
Passo inicial de ordenação de páginas em memória:

- pode ser usado qualquer algoritmo (e.g., quick sort)
- 13 leituras e 13 gravações de bloco ( $bd \cdot 2$  transferências)



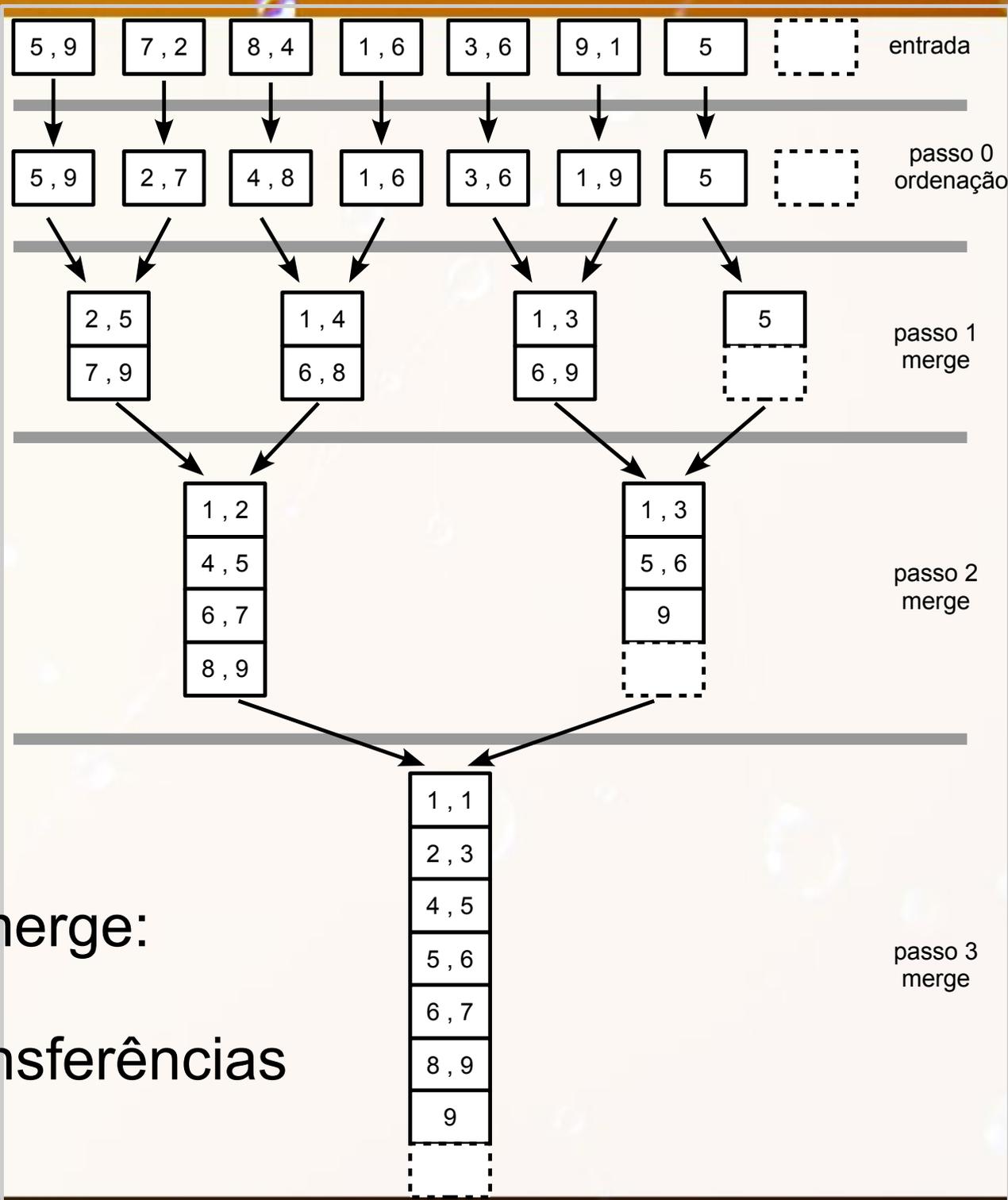
Primeiro merge:

- 3 blocos de memória (bm)
- 13 leituras e 13 gravações de bloco ( $bd \cdot 2$  transferências)



## Segundo merge:

- 3 blocos de memória (bm)
- 13 leituras e 13 gravações de bloco ( $bd \cdot 2$  transferências)



Terceiro merge:

- 3 bms
- $bd \cdot 2$  transferências

# Ordenação Externa

## Números

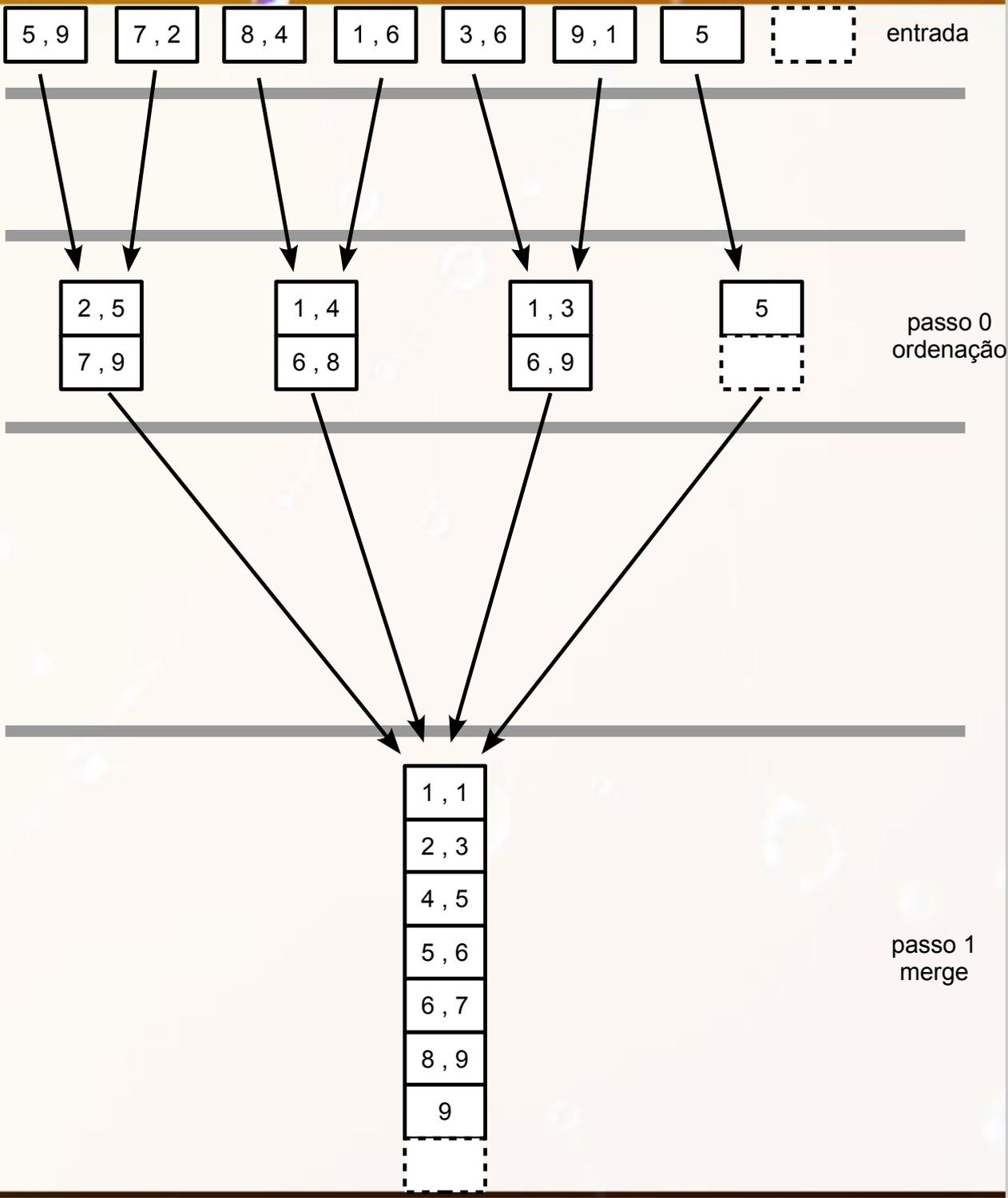
- bd - blocos em disco
- bm - blocos de memória
  - $bm_e$  - blocos de entrada =  $bm - 1$
  - $bm_s$  - blocos de saída = 1

# Ordenação Externa - Desempenho

- Ordenação - passo 0
  - $2*bd = 2*13 = 26$  transferências (leitura e gravação)
- Merge - a partir do passo 1
  - $2*bd = 2*13 = 26$  transferências a cada passo
- Rodadas por passo
  - $rodadas = \lceil bd/bm_e \rceil = \lceil 13/2 \rceil = 7$
- Total de Passos
  - $\lceil \log_2 rodadas \rceil + 1 = \lceil \log_2 7 \rceil + 1 = 4$  passos
- Custo:  $2*bd * (\lceil \log_2 rodadas \rceil + 1)$
- $\lceil \rightarrow$  notação para teto()

**Como Otimizar?**

**Se eu tiver 5 blocos de memória?**

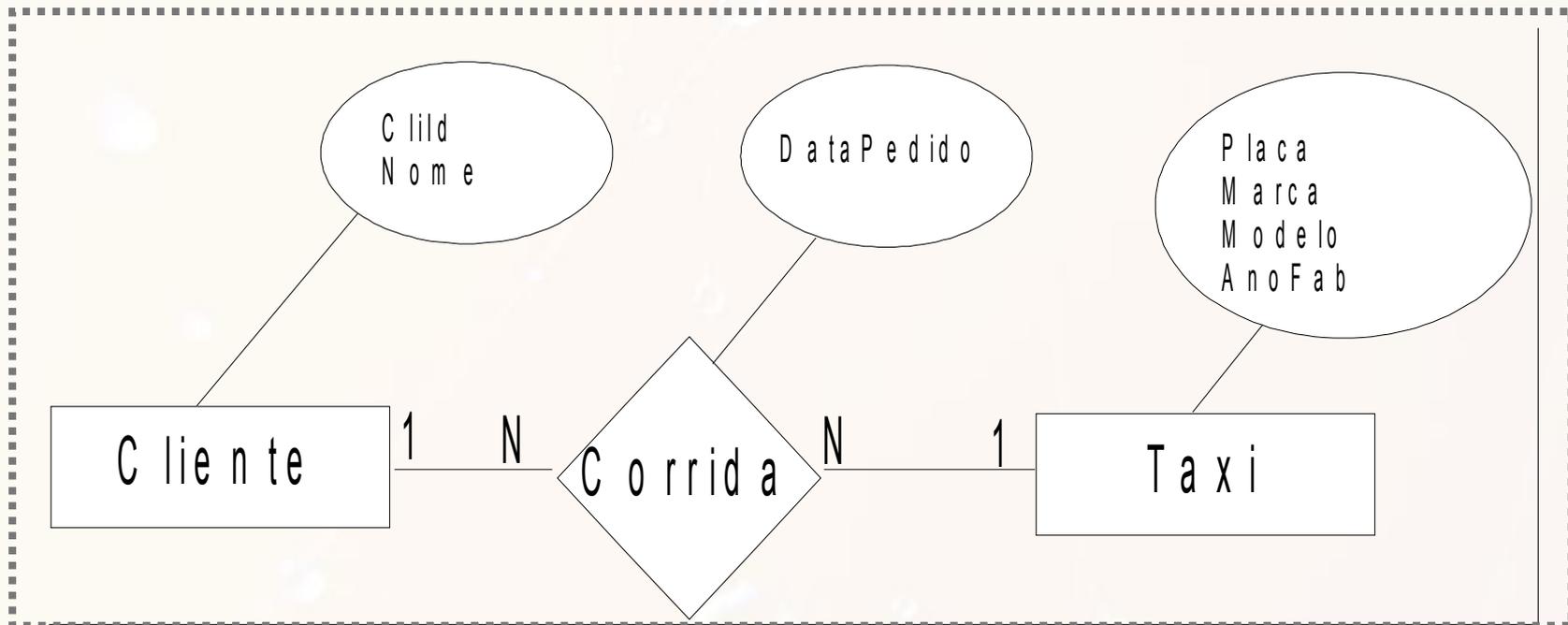


# Ordenação Externa - Desempenho

- Ordenação - passo 0
  - $2 * bd = 2 * 13 = 26$  transferências (leitura e gravação)
- Merge - a partir do passo 1
  - $2 * bd = 2 * 13 = 26$  transferências a cada passo
- Rodadas por passo
  - $rodadas = \lfloor bd / b_{m_e} \rfloor = \lfloor 13 / 4 \rfloor = 4$
- Total de passos
  - $\lfloor \log_{b_{m_e}} rodadas \rfloor + 1 = \lfloor \log_4 4 \rfloor + 1 = 2$  passos
- Custo:  $2 * bd * (\lfloor \log_{b_{m_e}} rodadas \rfloor + 1)$
- $\lfloor \rightarrow$  notação para teto()

# Seleção

# Esquema Conceitual - Exemplo Táxis



Este é um subconjunto do Estudo de Caso proposto “Despacho e controle de Táxis via terminais móveis ligados on-line com um sistema multi-usuário” por prof. Geovane Cayres Magalhães

# Tabelas para exemplo - Táxis

## Táxi (TX)

<u>Placa</u>	<u>Marca</u>	<u>Modelo</u>	<u>AnoFab</u>
D A E 6 5 3 4	F o r d	F i e s t a	1 9 9 9
D K L 4 5 9 8	W o l k s v a g e n	G o l	2 0 0 1
D K L 7 8 7 8	F o r d	F i e s t a	2 0 0 1
J D M 8 7 7 6	W o l k s v a g e n	S a n t a n a	2 0 0 2
J J M 3 6 9 2	C h e v r o l e t	C o r s a	1 9 9 9



## Corrida (R1)

<u>Ciid</u>	<u>Placa</u>	<u>DataPedido</u>
1 7 5 5	D A E 6 5 3 4	1 5 / 0 2 / 2 0 0 3
1 9 8 2	J D M 8 7 7 6	1 8 / 0 2 / 2 0 0 3



# Seleção?

$\sigma_{\text{Placa}='JDM8776'}(\text{TX})$

<u>Placa</u>	Marca	Modelo	Ano Fab
D A E 6 5 3 4	Ford	Fiesta	1 9 9 9
D K L 4 5 9 8	W o l k s v a g e n	G o l	2 0 0 1
D K L 7 8 7 8	Ford	Fiesta	2 0 0 1
J D M 8 7 7 6	W o l k s v a g e n	S a n t a n a	2 0 0 2
J J M 3 6 9 2	C h e v r o l e t	C o r s a	1 9 9 9

# Exatamente Igual Chave Primária

$\sigma_{\text{Placa}='JDM8776'}(\text{TX})$

<u>Placa</u>	Marca	Modelo	Ano Fab
D A E 6 5 3 4	Ford	Fiesta	1999
D K L 4 5 9 8	Wolksvagen	Gol	2001
D K L 7 8 7 8	Ford	Fiesta	2001
J D M 8 7 7 6	Wolksvagen	Santana	2002
J J M 3 6 9 2	Chevrolet	Corso	1999

# Exatamente Igual Outra Chave

$\sigma_{\text{AnoFab}=2002}(\text{TX})$

<u>Placa</u>	Marca	Modelo	AnoFab
D A E 6 5 3 4	Ford	Fiesta	1999
D K L 4 5 9 8	Wolksvagen	Gol	2001
D K L 7 8 7 8	Ford	Fiesta	2001
J D M 8 7 7 6	Wolksvagen	Santana	2002
J J M 3 6 9 2	Chevrolet	Corssa	1999

# Seleção?

$\sigma_{\text{AnoFab}=2002}(\text{TX})$

<u>Placa</u>	Marca	Modelo	Ano Fab
D A E 6 5 3 4	Ford	Fiesta	1 9 9 9
D K L 4 5 9 8	Wolkswagen	Gol	2 0 0 1
D K L 7 8 7 8	Ford	Fiesta	2 0 0 1
J D M 8 7 7 6	Wolkswagen	Santana	2 0 0 2
J J M 3 6 9 2	Chevrolet	Corso	1 9 9 9

# Seleção?

$\sigma_{\text{AnoFab} > 2000}(\text{TX})$

<u>Placa</u>	Marca	Modelo	Ano Fab
D A E 6 5 3 4	Ford	Fiesta	1 9 9 9
D K L 4 5 9 8	Wolkswagen	Gol	2 0 0 1
D K L 7 8 7 8	Ford	Fiesta	2 0 0 1
J D M 8 7 7 6	Wolkswagen	Santana	2 0 0 2
J J M 3 6 9 2	Chevrolet	Corso	1 9 9 9

# Faixa (>, <, >=, <=)

$\sigma_{\text{AnoFab}>2000}(\text{TX})$

<u>Placa</u>	Marca	Modelo	AnoFab
D A E 6 5 3 4	Ford	Fiesta	1999
D K L 4 5 9 8	Wolksvagen	Gol	2001
D K L 7 8 7 8	Ford	Fiesta	2001
J D M 8 7 7 6	Wolksvagen	Santana	2002
J J M 3 6 9 2	Chevrolet	Corso	1999

# Algoritmos de Seleção

- Exatamente igual
  - chave primária
  - outra chave
- $>$ ,  $<$ ,  $>=$ ,  $<=$
- compostos

# Algoritmos de Seleção

- Pesquisa linear
- Pesquisa binária
- Usando índice primário
- Usando chave hash
- Combinado com o índice primário
- Usando índice de agrupamento
- Usando índice secundário

# Seleção Conjuntiva x Dijuntiva

- seleção conjuntiva - e.g., and
- seleção dijuntiva - e.g., or

# Algoritmos de Seleção Conjuntiva

- Índice para uma das condições
- Índice composto envolvendo ambas as condições
- Índice individual para cada condição

# Seletividade

- seletividade: valor entre 0 e 1
- n registros
- igualdade atributo único
  - seletividade:  $1/n$

# Seletividade

## Atributo Não Único

- $i$  valores
- $i$  igualmente distribuído
- registros por valor?
- seletividade?

# Seletividade

## Atributo Não Único

- $i$  valores
- $i$  igualmente distribuído
- $n/i$  registros por valor
- seletividade:  $1/i$

# Seletividade

## Atributo Não Único

- primeiro as condições com valor menor de seletividade

# Exercício 1

- Considere a execução de uma consulta envolvendo uma seleção em um atributo que possui um índice. É sempre mais eficiente usar o índice do atributo no processamento?

# Junção (Join)

# Junção (Join) de Loop Aninhado

```
for each ti
  for each tj
    if match(ti, tj)
      add-result(ti, tj)
```

# Junção de Loop Aninhado Números

- $n_i$  - número de tuplas  $t_i$
- $n_j$  - número de tuplas  $t_j$
- pares de tuplas? (comparações?)

# Junção de Loop Aninhado

## Números

- $n_i$  - número de tuplas  $t_i$
- $n_j$  - número de tuplas  $t_j$
- $n_i * n_j$  - pares de tuplas

# Junção de Loop Aninhado

## Números

- $n_i$  - número de tuplas  $t_i$
- $n_j$  - número de tuplas  $t_j$
- $n_i * n_j$  - pares de tuplas
  
- $b_i$  - bloco de tuplas  $t_i$
- $b_j$  - bloco de tuplas  $t_j$
- leituras de blocos?

# Junção de Loop Aninhado

## Números

- $n_i$  - número de tuplas  $t_i$
- $n_j$  - número de tuplas  $t_j$
- $n_i * n_j$  - pares de tuplas
  
- $b_i$  - bloco de tuplas  $t_i$
- $b_j$  - bloco de tuplas  $t_j$
- $b_i + b_j * n_i$  leituras de blocos

# Junção de Loop Aninhado

## Números

- Situações:
  - Quantas transferências de bloco se todos os blocos estiverem na memória?
  - Quantas transferências se os blocos de um dos loops estiver todo na memória e qual deles escolher (bi ou bj)

# Junção de Loop Aninhado

## Números

### ■ Situações:

- Quantas transferências de bloco se todos os blocos estiverem na memória?
  - $b_i + b_j$  transferências
- Quantas transferências se os blocos de um dos loops estiver todo na memória e qual deles escolher ( $b_i$  ou  $b_j$ )?
  - escolher  $b_j$
  - $b_i + b_j$  transferências

# Junção de Loop Aninhado em Bloco

```
for each bi
  for each bj
    for each ti in bi
      for each tj in bj
        if match(ti, tj)
          add-result(ti, tj)
```

# Junção de Loop Aninhado em Bloco Números

- $b_i$  - bloco de tuplas  $t_i$
- $b_j$  - bloco de tuplas  $t_j$
- leituras de blocos?

# Junção de Loop Aninhado em Bloco Números

- $b_i$  - bloco de tuplas  $t_i$
- $b_j$  - bloco de tuplas  $t_j$
- $b_i + b_j * b_i$  leituras de blocos

# Exercício 2

- Considere as seguintes tabelas e consulta:
  - Aluno (ra, nome, id\_dept)
  - Departamento (id\_dept, nome\_dept)
  - ```
SELECT ra, nome, nome_dept
FROM Aluno, Departamento
WHERE Aluno.id_dept = Departamento.id_dept
```
- Escreva o pseudo-código para o processamento do join na consulta acima.
  - a) Considere que todas as tabelas cabem na memória.
  - b) Considere que apenas a tabela Departamento cabe na memória.

# Outras Junções

- Junção Indexada
- Junção Merge
- Junção Hash

# Projeção

- Recorte dos campos
- (?)

# Projeção

- Recorte dos campos
- Registros sem duplicatas
  - SQL → padrão não eliminar duplicatas
    - DISTINCT → elimina duplicatas
  - Registros com garantia de ser únicos
    - e.g., contendo chave primária
  - Registros sem garantia de ser únicos
    - ordenação
    - hashing

# Otimização de Consulta

# SQL p/ Álgebra

## ■ Versão SQL

```
SELECT Codigo, Nome  
FROM PESSOA  
WHERE AnoFiliacao = 1990
```

## ■ Versão em álgebra

$$\pi_{\text{Codigo, Nome}}(\sigma_{\text{AnoFiliacao}=1990}(\text{PESSOA}))$$

## ■ Versão Árvore



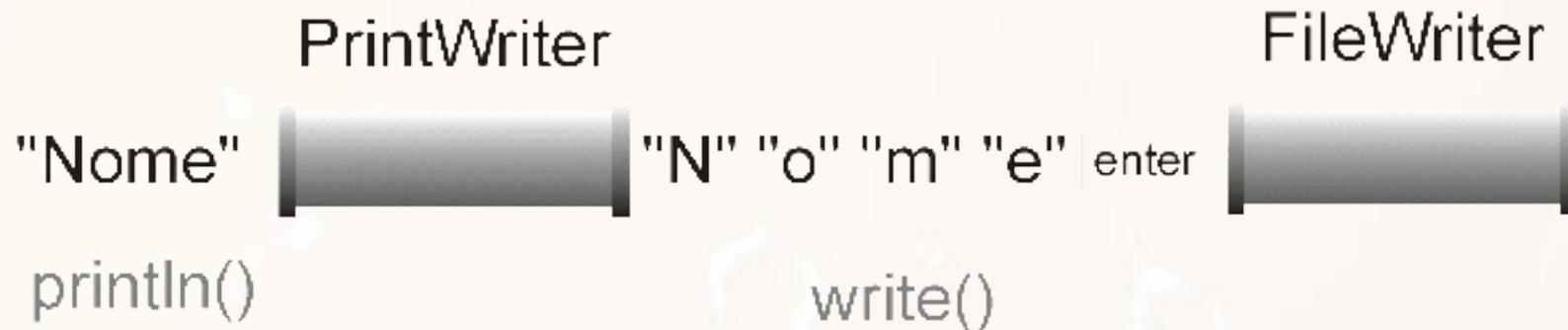
# Combinação de Operações usando Pipelining

- Uma consulta é mapeada em uma sequência de operações
- A execução de cada operação produz um resultado temporário
- Alternativa
  - Evitar ao máximo resultados temporários
  - Pipelining
    - concatena operações
    - conforme uma saída é produzida gera entrada para a operação subsequente

# Pipelining Pattern Pipe & Filter



**exemplo: Java Writer**



# Exemplo de Pipeline

## P E S S O A

| <u>C o d i g o</u> | N o m e         | T e l e f o n e   | A n o F i l i a c a o |
|--------------------|-----------------|-------------------|-----------------------|
| 1 5 2 5            | A s d r ú b a l | 5 4 3 2 - 1 0 9 8 | 1 9 9 0               |
| 1 6 3 7            | D o r i a n a   | 9 8 7 6 - 5 4 3 2 | 1 9 8 3               |
| 1 7 0 1            | Q u i n c a s   | 8 7 6 5 - 4 3 2 1 | 1 9 8 5               |
| 2 0 4 2            | M e l i s s a   | 7 6 5 4 - 3 2 1 0 | 1 9 9 0               |
| 2 1 1 1            | H o r á c i o   | 6 5 4 3 - 2 1 0 9 | 1 9 8 3               |

P E S S O A

# Exemplo de Pipeline

$\sigma$  AnoFiliacao = 1990

P E S S O A

## P E S S O A

| <u>C o d i g o</u> | N o m e         | T e l e f o n e   | A n o F i l i a c a o |
|--------------------|-----------------|-------------------|-----------------------|
| 1 5 2 5            | A s d r ú b a l | 5 4 3 2 - 1 0 9 8 | 1 9 9 0               |
| 1 6 3 7            | D o r i a n a   | 9 8 7 6 - 5 4 3 2 | 1 9 8 3               |
| 1 7 0 1            | Q u i n c a s   | 8 7 6 5 - 4 3 2 1 | 1 9 8 5               |
| 2 0 4 2            | M e l i s s a   | 7 6 5 4 - 3 2 1 0 | 1 9 9 0               |
| 2 1 1 1            | H o r á c i o   | 6 5 4 3 - 2 1 0 9 | 1 9 8 3               |

# Exemplo de Pipeline

## P E S S O A

| <u>C o d i g o</u> | N o m e         | T e l e f o n e   | A n o F i l i a c a o |
|--------------------|-----------------|-------------------|-----------------------|
| 1 5 2 5            | A s d r ú b a l | 5 4 3 2 - 1 0 9 8 | 1 9 9 0               |
| 2 0 4 2            | M e l i s s a   | 7 6 5 4 - 3 2 1 0 | 1 9 9 0               |

$\sigma$  A n o F i l i a c a o = 1 9 9 0



# Exemplo de Pipeline



PESSOA

| <u>C od i g o</u> | N o m e         | T e l e f o n e   | A n o F i l i a c a o |
|-------------------|-----------------|-------------------|-----------------------|
| 1 5 2 5           | A s d r ú b a l | 5 4 3 2 - 1 0 9 8 | 1 9 9 0               |
| 2 0 4 2           | M e l i s s a   | 7 6 5 4 - 3 2 1 0 | 1 9 9 0               |

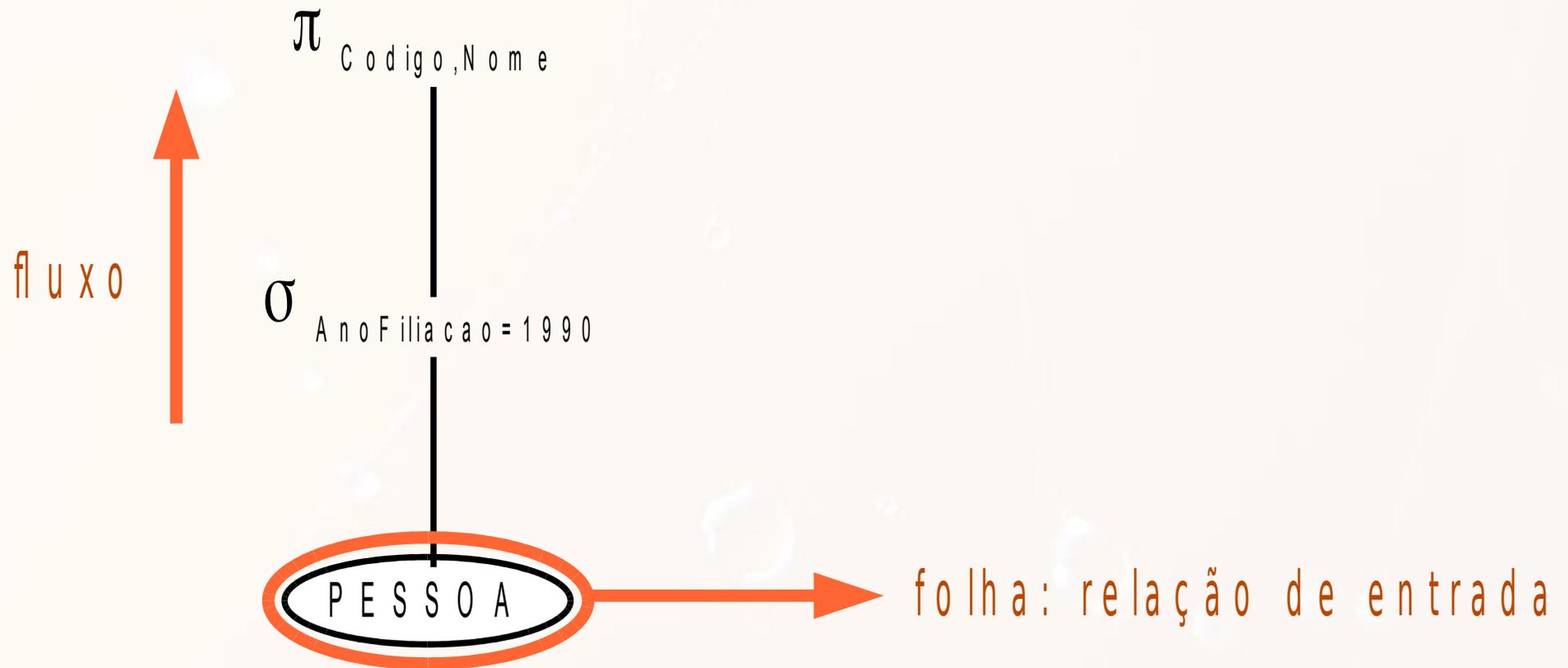
# Exemplo de Pipeline



P E S S O A

| <u>C o d i g o</u> | N o m e         |
|--------------------|-----------------|
| 1 5 2 5            | A s d r ú b a l |
| 2 0 4 2            | M e l i s s a   |

# Árvore de Consulta



# Heurísticas para Otimização de Consulta (Elmasri, 2011)

# Heurísticas para Otimização de Consulta

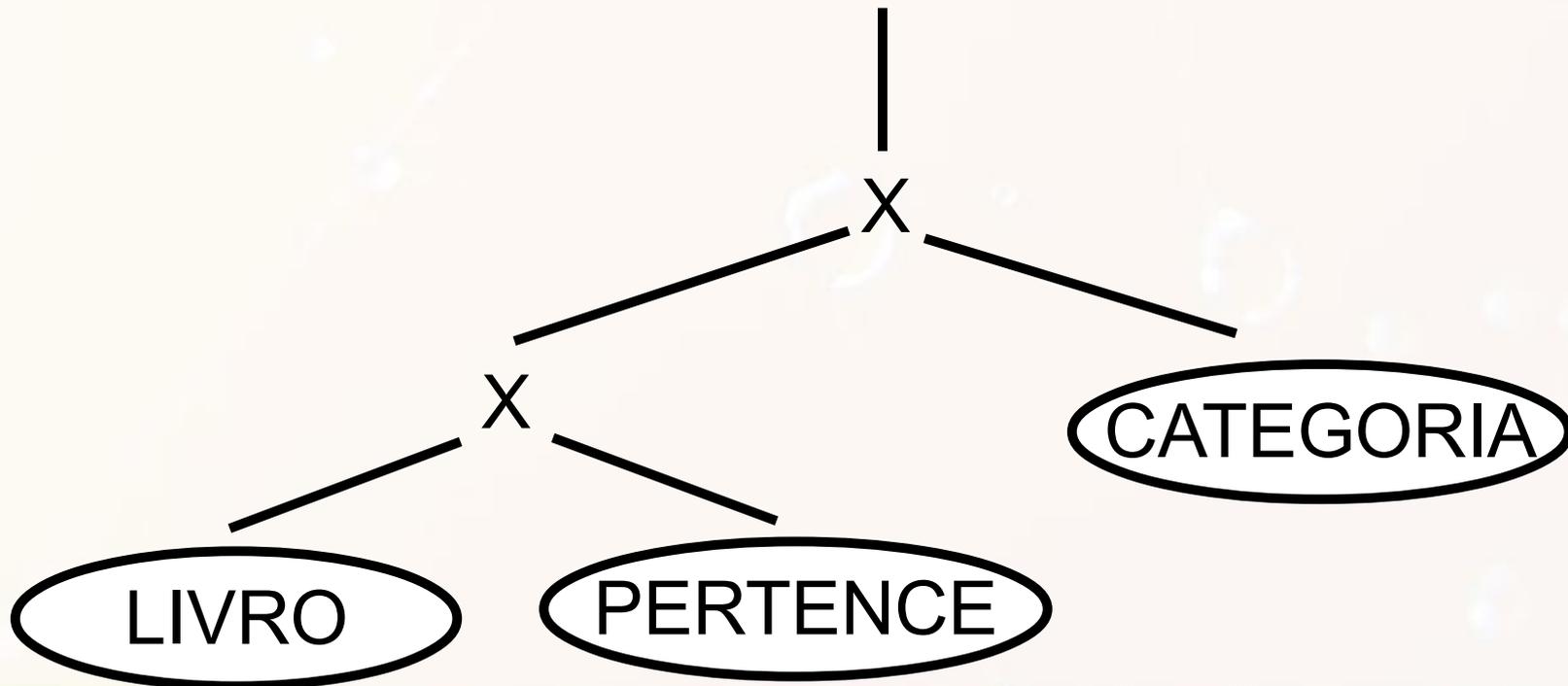
- Título dos livros sobre poesia escritos depois de 1996

```
SELECT LIVRO.Titulo
FROM LIVRO, PERTENCE, CATEGORIA
WHERE CATEGORIA.Nome = "poesia" AND
LIVRO.ISBN = PERTENCE.ISBN AND
CATEGORIA.Codigo = PERTENCE.CodCategoria AND
LIVRO.Ano > 1996
```

# Heurística para Otimização de Consulta

$\pi_{\text{LIVRO.Titulo}}$

$\sigma_{\text{CATEGORIA.Nome="poesia" AND LIVRO.ISBN=PERTENCE.ISBN AND CATEGORIA.Codigo=PERTENCE.CodCategoria AND LIVRO.Ano>1996}}$



# Regras de Transformação

1. Operações seleção conjuntivas podem se converter em cascatas de seleção
2. Operação de seleção é comutativa
3. Comutação de seleção com projeção
  - caso o resultado da projeção tenha atributos requeridos pela seleção

# Regras de Transformação

4. Seleção e junção (ou produto cartesiano) são comutativas

- se atributos da seleção são de apenas uma das relações

5. Operações de união e interseção são comutativas

- diferença não é

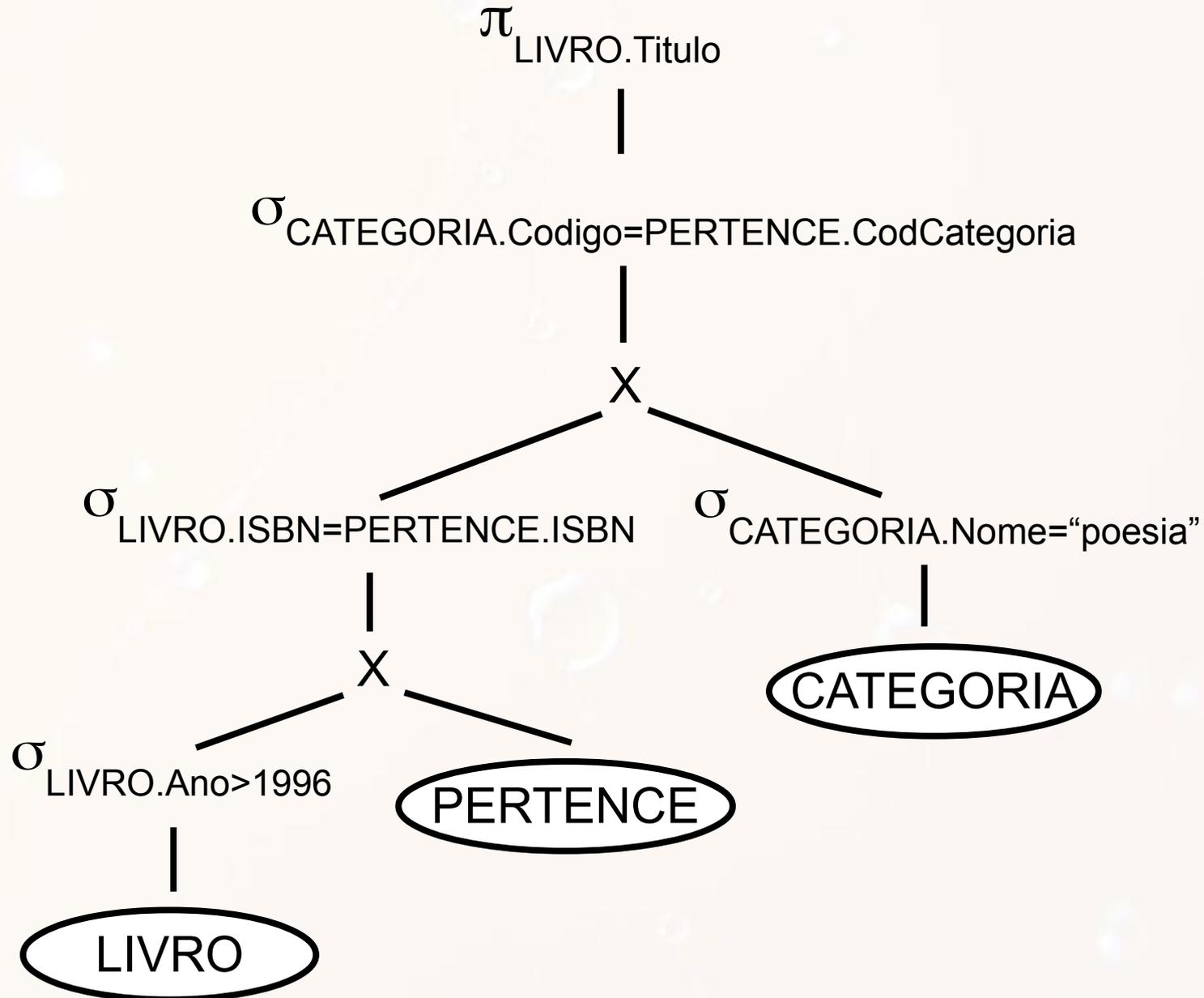
# Regras de Transformação

6. Seleção é comutativa com operações de conjunto (união, interseção e diferença)
- $\text{sel}(A @ B)$  equivale  $\text{sel}(A) @ \text{sel}(B)$

# Heurísticas

- Quebrar operações de seleção conjuntivas (1)
  - maior liberdade
- Mover seleção em direção às folhas (2), (3), (4), (5) e (6)
  - apenas 1 tabela → acima da tabela
  - duas tabelas → acima da junção

# Quebrando e Descendo Seleções



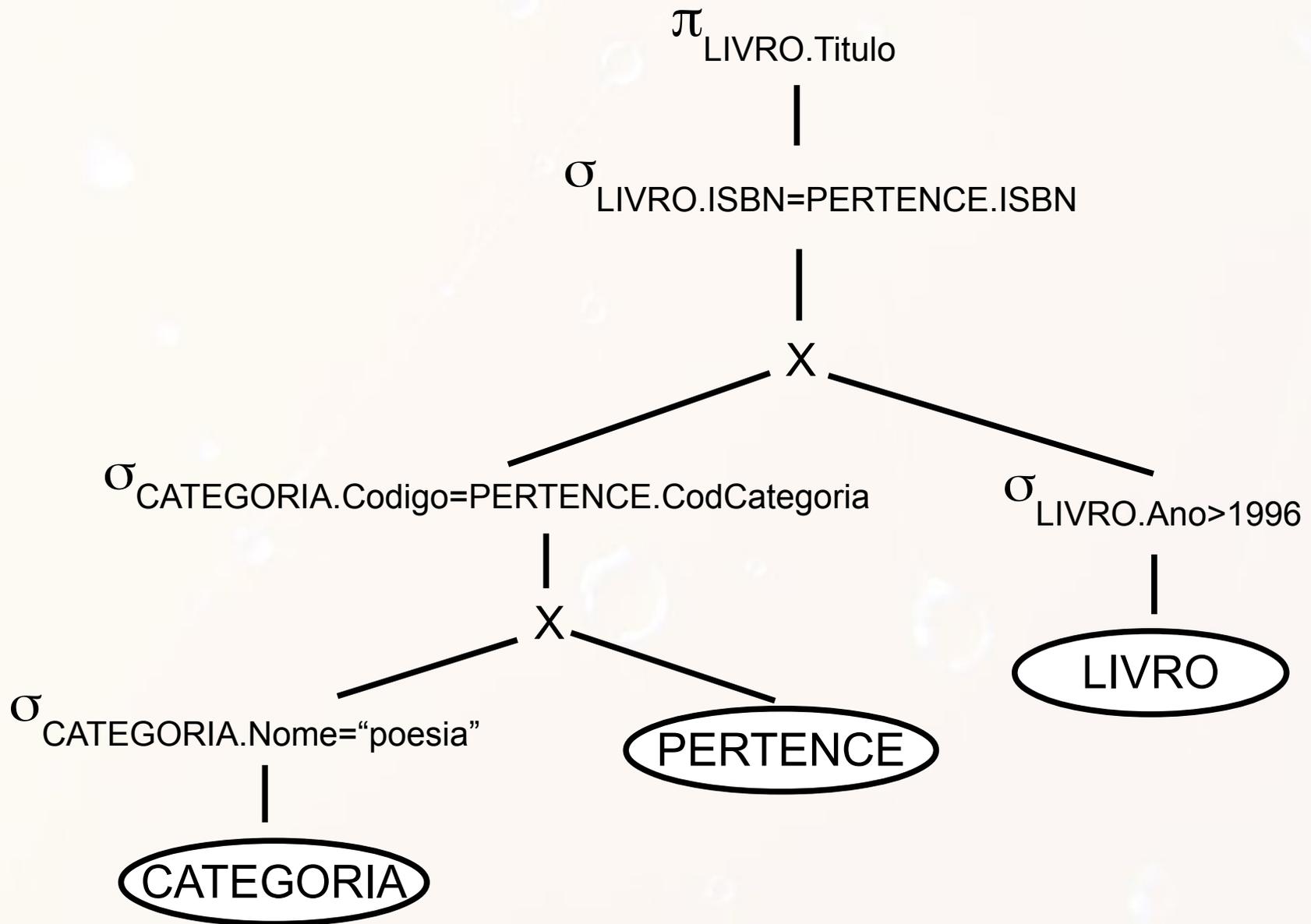
# Regras de Transformação

7. As operações de junção e produto cartesiano são comutativas
8. As operações de junção, produto cartesiano, união e interseção são associativas

# Heurística

- Operações de seleção mais restritivas devem ser executadas primeiro (5) e (6)

# Troca de Categoria com Livro



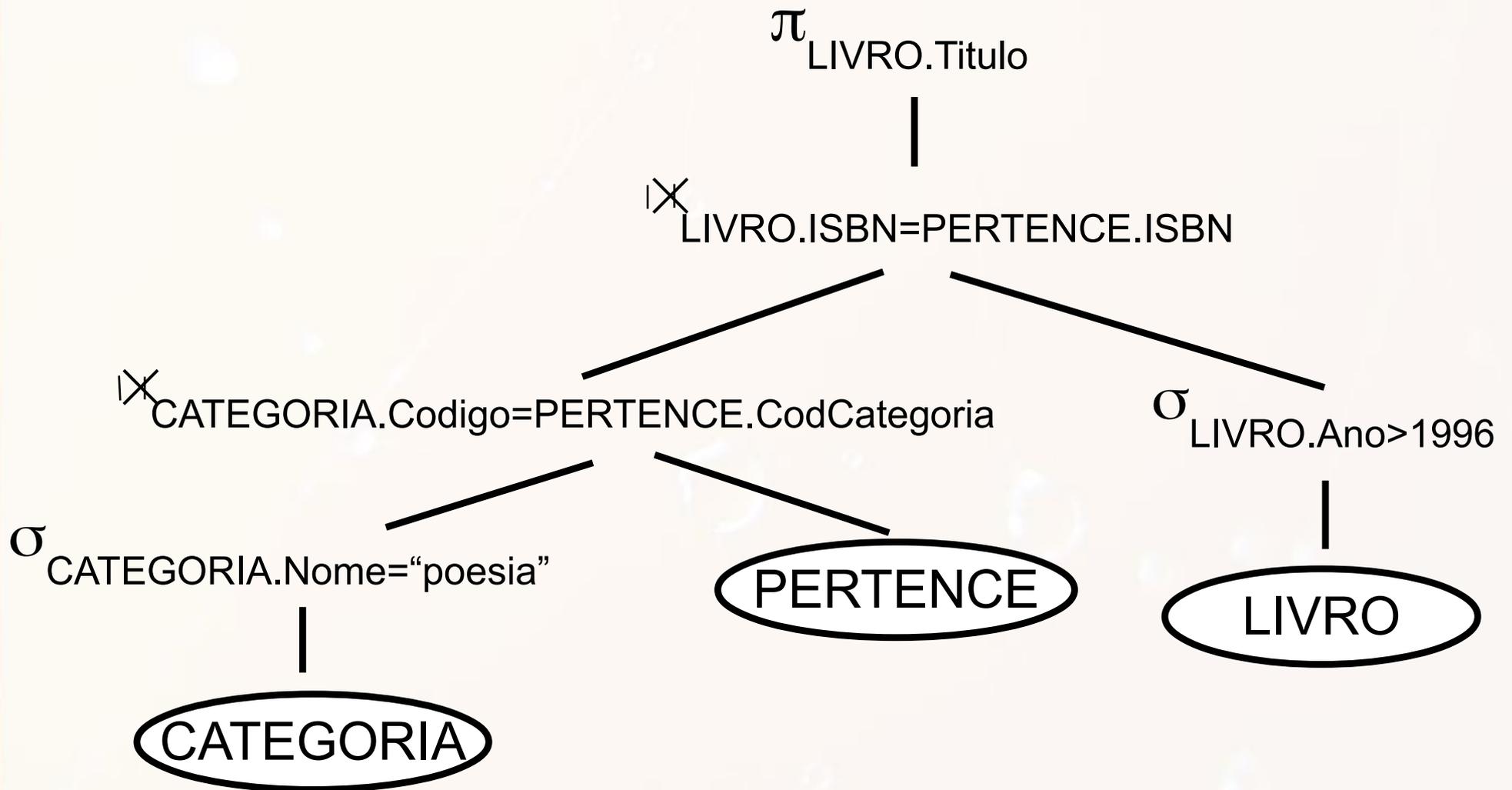
# Regra de Transformação

9. Operações de produto cartesiano + seleção podem se converter em junção

# Heurística

- Converta produtos cartesianos + seleções em junções

# Produto Cartesiano + Seleção = Junção



# Regras de Transformação

10. Cascata de projeções podem ser ignoradas e convertidas na última

- $\text{Pr1}(\text{Pr2}(\text{Pr3}(A)))$  equivale  $\text{Pr1}(A)$

11. Operações de projeção e união são comutativas

- $\text{proj}(A \cup B)$  equivale  $\text{proj}(A) \cup \text{proj}(B)$

# Regras de Transformação

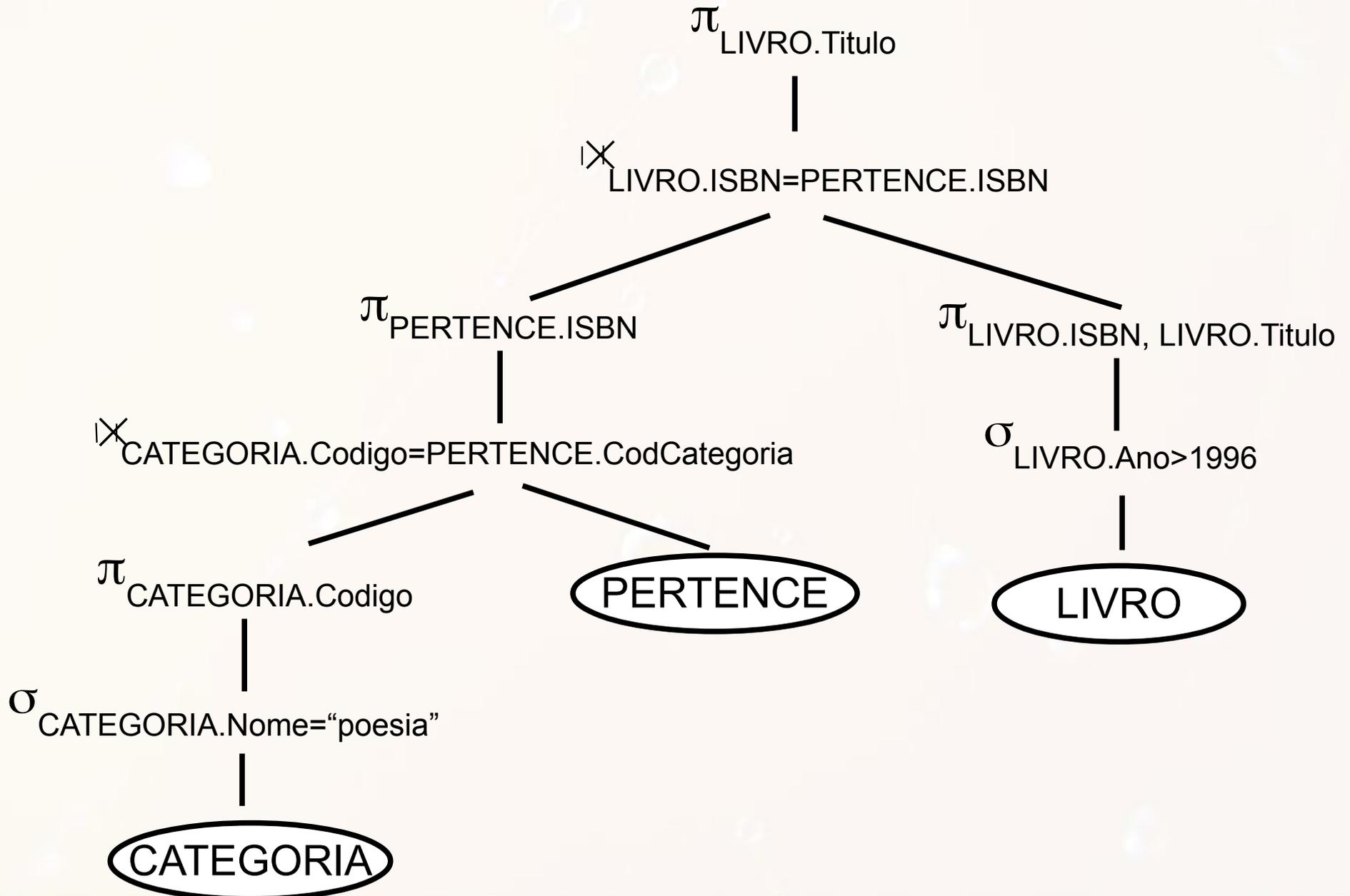
## 12. Operação de projeção pode ser comutada com junção (ou produto cartesiano)

- Relação  $A \rightarrow$  atributos  $a_1, \dots, a_n$
- Relação  $B \rightarrow$  atributos  $b_1, \dots, b_m$
- $L = (a_1, \dots, a_n, b_1, \dots, b_m)$
- Condição só contém atributos  $L$
- $\text{proj}_L(A \text{ junção } B)$  equivale  $(\text{proj}_{a_1, \dots, a_n}(A)) \text{ junção } (\text{proj}_{b_1, \dots, b_m}(B))$

# Heurística

- Baseados em (10), (11) e (12)
  - Desmembrar operações de projeção
  - Mover projeções em direção às folhas
  - Criar operações de projeção para manter apenas atributos necessários

# Projeções Mais Cedo



# Heurística

- Identificar subárvores com operações a ser combinadas em um algoritmo

# Exercício 3

- Considere as seguintes tabelas:

- $R(\underline{A}, B, C, D)$

- $S(\underline{E}, F, G, H)$  - E é chave-estrangeira que referencia R(A)

a) desenhe um plano de acesso otimizado para a consulta:

- `select A from R, S  
where A=5 and G=7 and F=A`

# Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides. Página do Celso: <http://dainf.ct.utfpr.edu.br/~gomesjr/>
- Patrícia Cavoto (professora desta disciplina em 2015) pela contribuição na disciplina e nos slides.

# Referências

- Elmasri, Ramez; Navathe, Shamkant B. (2005) **Sistemas de Bancos de Dados**. Addison-Wesley, 4<sup>a</sup> edição em português.
- Elmasri, Ramez; Navathe, Shamkant B. (2011) **Sistemas de Bancos de Dados**. Addison-Wesley, 6<sup>a</sup> edição em português.
- Ramakrishnan, Raghu; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3<sup>rd</sup> edition.

**André Santanchè**

<http://www.ic.unicamp.br/~santanche>

# Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.
- Mais detalhes sobre a referida licença Creative Commons veja no link:  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>
- Fotografia da capa e fundo por  
<http://www.flickr.com/photos/fdecomite/>  
Ver licença específica em  
<http://www.flickr.com/photos/fdecomite/1457493536/>