

Implementação de operadores relacionais

BDII

Introdução

- Algumas operações são caras!
- Podemos melhorar a performance melhorando a operação
 - Podemos melhorar até 1.000.000x
 - Principais armas são:
 - Boas implementações de operadores
 - Explorar equivalência de operadores
 - Usar estatísticas para estimar o custo de uma operação

Esquema Geral do Otimizador

Consulta SQL



SQL Parser

Otimizador

Melhor Plano de execução

Bloco SQL simples

Transforma em Álgebra

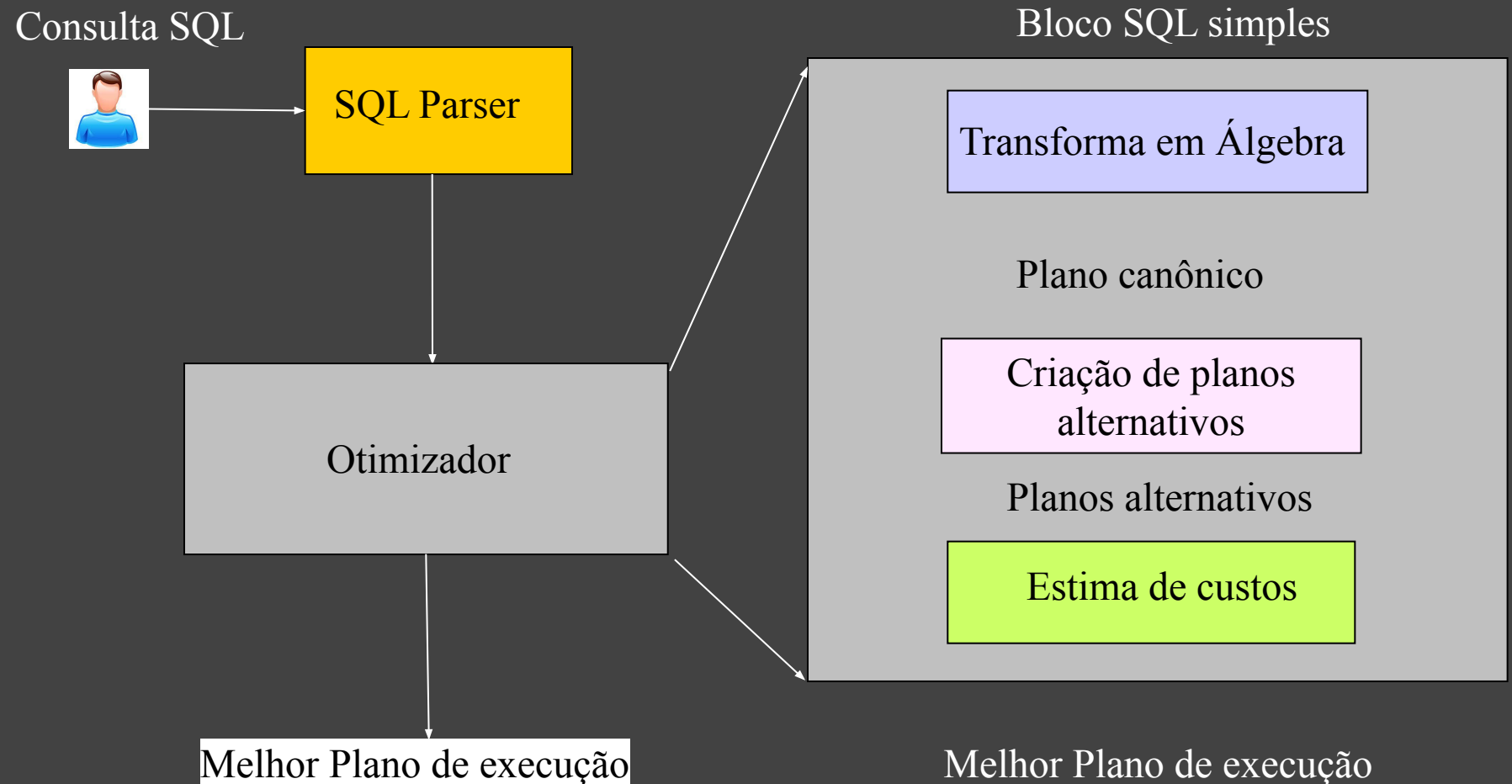
Plano canônico

Criação de planos
alternativos

Planos alternativos

Estima de custos

Melhor Plano de execução



Catálogo do Sistema

- Informações armazenadas no catálogo, necessárias no processo de otimização:
- Informações gerais:
 - Tamanho do buffer pool (espaço livre)
 - Tamanho de uma página em disco
- Informações sobre as tabelas
- Informações sobre índices
- Estatísticas sobre tabelas e índices: atualizadas periodicamente

Estatísticas sobre tabelas e índices:

- $NTuples(R)$ = Número de tuplas da tabela R
- $NPages(R)$ = Número de páginas da tabela R
- $NKeys(I)$ = número de chaves distintas do índice
- $INPages(I)$ = número de páginas do índice
- $IHeight(I)$ = Altura do Índice (no caso de B+tree)
- $ILow(I)$ = menor valor de chave do índice I
- $IHigh(I)$ = maior valor de chave do índice I

Esquema

Marinheiros (*sid*: integer, *sname*: char, *rating*: integer)

Reservas (*sid*: integer, *bid*: integer, *day*: dates)

- Reservas:
 - Tupla com 40 bytes, 100 tuplas por página, 1000 páginas.
- Marinheiros:
 - Tupla 50 bytes, 80 tuplas por página, 500 páginas.

Exemplo

Marinheiros

sid	sname	dating	Age
22	Joao	9	45
28	Pedro	8	35
31	Maria	3	25

Reservas

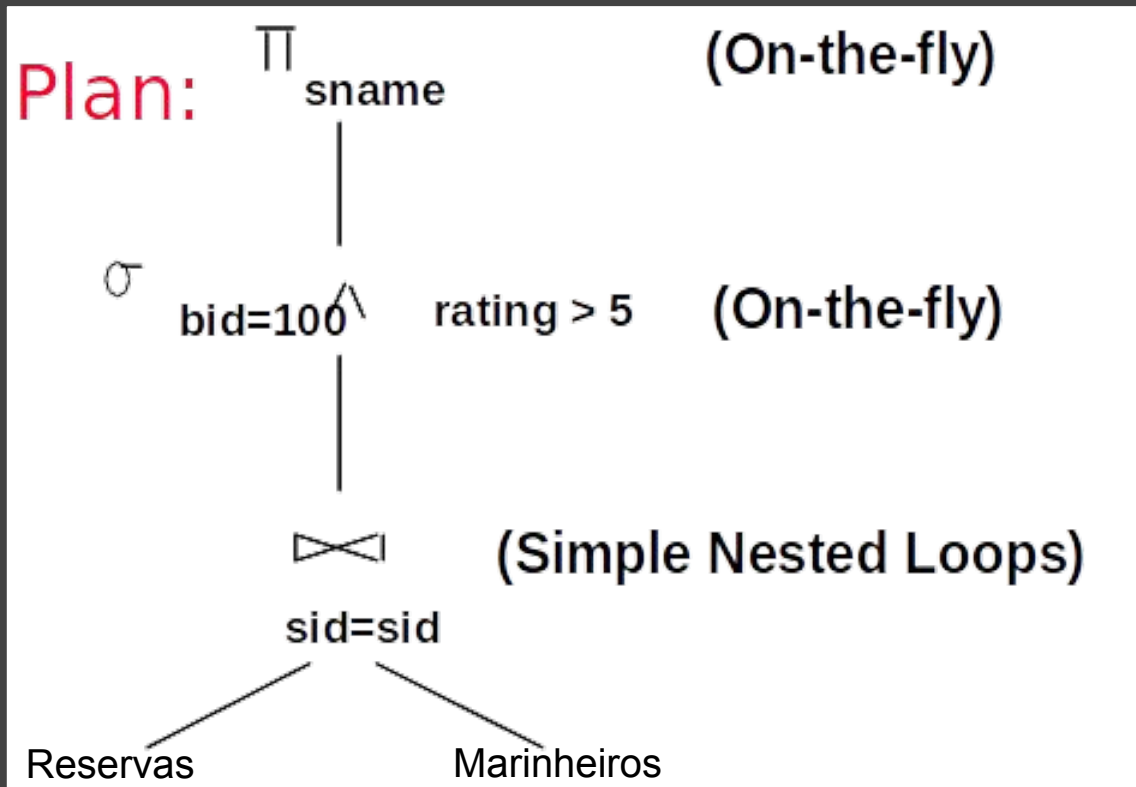
sid	bid	day
28	103	01/02/2022
28	103	03/1/2021
31	101	23/02/2019
31	102	30/05/2023
31	101	01/07/2015

Plano de consulta

```
SELECT S.sname  
FROM Reservas R, Marinheiros S  
WHERE R.sid=S.sid AND  
       R.bid=101 AND S.rating>5
```


Plano de consulta

```
SELECT S.sname
FROM Reservas R, Marinheiros S
WHERE R.sid=S.sid AND
      R.bid=100 AND S.rating>5
```



Operadores relacionais

- Vamos considerar os seguintes operadores:
 - Seleção (σ)
 - Projeção (π)
 - Join (\bowtie)

Joins

- Joins são comuns em SGBDs
- Joins são bastante custosos
- Muitas abordagens para reduzir o custo

Page (or block)-Oriented Nested Loops Join

```
foreach page  $b_R$  in R do
  foreach page  $b_m$  in M do
    foreach tuple  $r$  in  $b_R$  do
      foreach tuple  $m$  in  $b_m$  do
        if  $r_i == m_j$  then add  $\langle r, m \rangle$  to result
```

- Para cada página de R, buscar uma página de S e escrever tuplas $\langle r, m \rangle$

R=1000 pag

S=500 pag

Custo: $R*S + R$

Custo = $1000*500 + 1000=501.000$ I/O

Page (or block)-Oriented Nested Loops Join

Marinheiros

Sid	Sname	Rating	age
22	Joao	9	45
28	Pedro	8	35
31	Maria	3	25

Reservas

Sid	Bid	Day
28	103	01/02/2022
28	103	03/1/2021
31	101	23/02/2019
31	102	30/05/2023
31	101	01/07/2015

Páginas I/O:

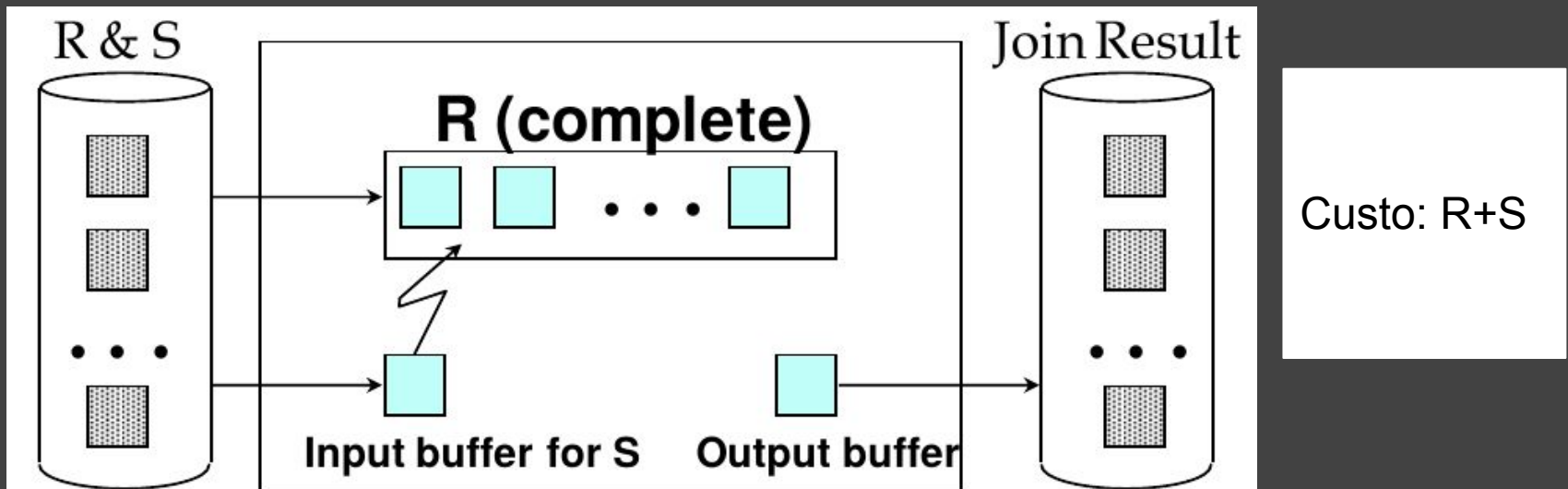
***OBS: neste exemplo uma tupla ocupa uma página na memória

Page (or block)-Oriented Nested Loops Join (PON)

Aplicando o Join com o algoritmo PON em uma relação com 100 registros e outra com 1000 registros. Qual o custo de I/O sabendo que 1 registro ocupa uma página no disco?

Simple nested loops join

Objetivo: Carregar a relação menor inteira na memória



Simple nested loops join: exemplo

Marinheiros

Sid	Sname	Rating	age
22	Joao	9	45
28	Pedro	8	35
31	Maria	3	25

Reservas

Sid	Bid	Day
28	103	01/02/2022
28	103	03/1/2021
31	101	23/02/2019
31	102	30/05/2023
31	101	01/07/2015

Custo:

Páginas I/O:

***OBS: neste exemplo uma tupla ocupa uma página na memória

Simple Nested Loops join

Marinheiros

Sid	Sname	Rating	age
22	Joao	9	45
28	Pedro	8	35
31	Maria	3	25

Reservas

Sid	Bid	Day
28	103	01/02/2022
28	103	03/1/2021
31	101	23/02/2019

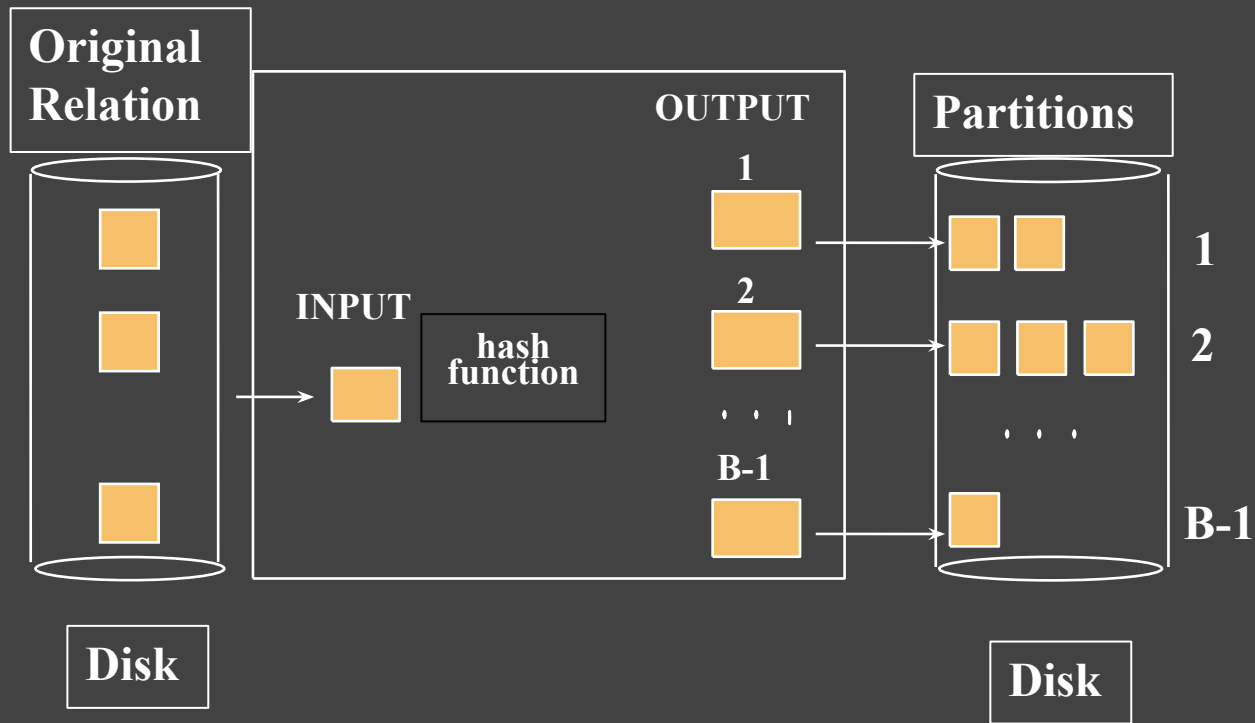
Qual seria o custo de se aplicar o algoritmo Simple Nested Loop join nas tabelas acima? Considere que cada registro é armazenado em uma página. Simule a execução do algoritmo. Considere que a memória é capaz de armazenar 6 páginas.

Simple nested loops join

Aplicando o Join com o algoritmo Simple Nested Loops Join em uma relação com 100 registros e outra com 1000 registros. Qual o custo de i/o sabendo que 4 registros ocupam uma página no disco?

Memória com 26 páginas

Hash-Join



Hash-Join

Marinheiros

Sid	Sname	Rating	age
22	Joao	9	45
28	Pedro	8	35
31	Maria	3	25
40	José	8	60

0		
1		
2		
3		

Reservas

Sid	Bid	Day
28	103	01/02/2022
28	103	03/1/2021
31	101	23/02/2019
31	102	30/05/2023
31	101	01/07/2015

0			
1			
2			
3			

Atividade - Hash-Join

Marinheiros

Sid	Sname	Rating	age
20	Joao	9	45
21	Pedro	8	35
22	Maria	3	25

Reservas

Sid	Bid	Day
20	103	01/02/2022
20	103	03/1/2021
22	101	23/02/2019
30	102	30/05/2023
45	101	01/07/2015
22	101	30/05/2023

Custo do Hash-Join

Fase de partição: ler e escrever ambas $2(M+N)$

Fase de matching: ler ambas $(M+N)$

Total: $3(M+N)$

Merge-Sort Join ($R \bowtie_{i=j} S$)

- Ordenar R e S na coluna comum, então realizar um merge e retornar o resultado
- Importante se:
 - Uma ou ambas as entradas já estão ordenadas;
 - A saída deve ser ordenada;
- Algoritmo:
 - Fase de ordenamento: ordenar ambas relações usando sort (externo se precisar)
 - Fase de junção: verificar as tuplas qualificadas com a junção de ambas as tabelas;

Merge-Sort

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

<u>sid</u>	<u>bid</u>	<u>day</u>	rname
28	103	12/4/96	guppy
28	103	11/3/96	yuppy
31	101	10/10/96	dustin
31	102	10/12/96	lubber
31	101	10/11/96	lubber
58	103	11/12/96	dustin

Index Nested Loops Join

```
foreach tuple r in R do
  foreach tuple s(INDEXADA) in S where  $r_i == s_j$  do
    add <r, s> to result
```

Se existir um índice sobre uma das relações (ex. S):

Cost: $R + (R * p_R) * \text{custo de encontrar a matching tuple S}$

Para cada tupla de R, o custo de indexar S é de 2-4 IOs para B+

Atividade

Uma relação R (com 150 páginas) consiste de um att A e outra relação S (com 200 pág) também contém o att A. Determinar o melhor método de join para a query:

```
select * from R, S where R.A = S.A
```

Assumir que: buffer=10 pag; não existem índices; e os seguintes métodos de join estão implementados: advanced block nested loop and hash-join. **Determinar o número de I/Os necessárias em cada método, se possível.**

- Simple Nested-loop:

- Hash - join