# TRABALHO PRÁTICO 3

# Código Fonte

- O SGBD criado para a disciplina suporta dois algoritmos de junção
  - MergeJoin
    - Exige que as tabelas estejam ordenadas
  - NestedLoopJoin
    - Não exige que as tabelas estejam ordenadas

# Algoritmos de junção

- No exemplo abaixo,
  - Os registros de tab1 e tab2 estão divididos em blocos
  - a junção entre tab1 e tab2 encontra apenas três correspondências

Tak	<b>o</b> 1
-----	------------

4	Joao
7	Ana
3	Maria

1	Pedro

|X| Tab 2

1	Rua do beco	
2	Sem rua	
3	3 Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

#### Resposta

4	Joao – Avenida Brasil	
3	Maria - Subúrbio	
1 Pedro - Rua do Beco		

# **NESTED LOOP JOIN**

- O NestedLoopJoin (NLJ) tem um custo em número de transferências de blocos proporcional à seguinte equação
  - Br+ Nr \* Bs

#### Onde

- Br = número de blocos da relação externa
- Nr = número de registros da relação externa
- Bs = número de blocos da relação interna

- O exemplo a seguir mostra como funcionaria a varredura das tabelas usando o NLJ
- O algoritmo assume que um registro de um lado pode ter correspondências com mais do que um registro do outro lado
  - Ou seja, ele não para de procurar quando uma correspondência for encontrada

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

#### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 0

O cursor em tab1 é aberto e vai para primeiro registro

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

Tab 2 (S)



1	Rua do beco	
2	Sem rua	
3	3 Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

O cursor em tab2 é aberto e vai para o primeiro registro

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

### Tab 2 (S)



1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil
5	Rio Guaíba

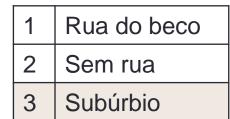
#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro
	_

#### Tab 2 (S)





4	Avenida Brasil
5	Rio Guaíba

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

#### Tab1 (R)

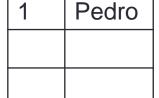


4	Joao
7	Ana
3	Maria

4, Joao - Avenida Brasil

#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio





4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Correspondência encontrada . Registro é retornado

#### Tab1 (R)

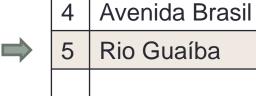


4	Joao
7	Ana
3	Maria

1	Pedro

### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Tab2 avança

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	



#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro
	_

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

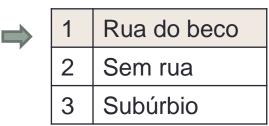
Tab1 avança

#### Tab1 (R)

	4	Joao
$\Rightarrow$	7	Ana
	3	Maria

1	Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 3

O cursor em tab2 é aberto e vai para o primeiro registro

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro

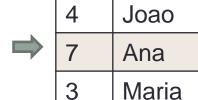
### Tab 2 (S)



1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

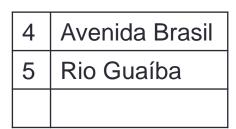
#### Tab1 (R)



1	Pedro

### Tab 2 (S)

	1	Rua do beco	
	2	Sem rua	
•	3	Subúrbio	



#### Tab1 (R)



1	Pedro

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 4

Tab2 avança

#### Tab1 (R)

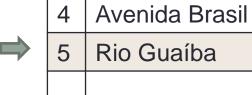


4	Joao
7	Ana
3	Maria

1	Pedro
	_

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 4

Tab2 avança

#### Tab1 (R)



4	Joao
7	Ana
3	Maria

1	Pedro
	_

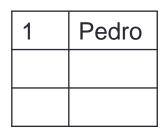
### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

	4	Avenida Brasil
	5	Rio Guaíba
$\Rightarrow$		

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

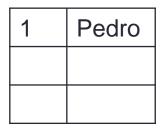
4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 4

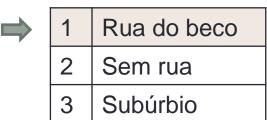
Tab1 avança

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



#### Tab 2 (S)



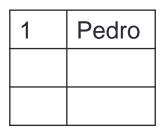
4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 5

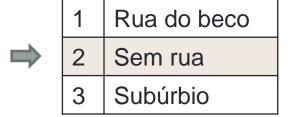
O cursor em tab2 é aberto e vai para o primeiro registro

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



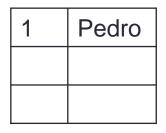
### Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

### 3, Maria – Subúrbio



1	Rua do beco
2	Sem rua
3	Subúrbio

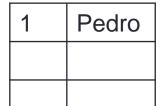


4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 5

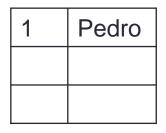
Correspondência encontrada. Registro é retornado





#### Tab1 (R)

4	Joao
7	Ana
3	Maria



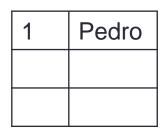
### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



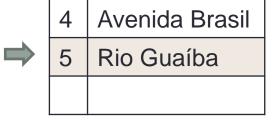
#### Tab1 (R)

4	Joao
7	Ana
3	Maria



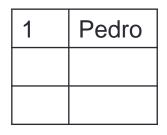
### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



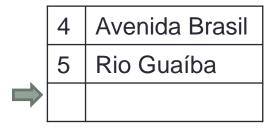
#### Tab1 (R)

4	Joao
7	Ana
3	Maria



### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



#### Tab1 (R)

4	Joao
7	Ana
3	Maria



1	Pedro

### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 6

Tab1 avança

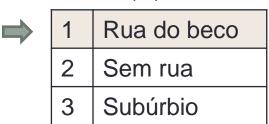
#### Tab1 (R)

4	Joao
7	Ana
3	Maria



1	Pedro

#### Tab 2 (S)



4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 7

O cursor em tab2 é aberto e vai para o primeiro registro

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1, Pedro – Rua do Beco

#### Tab 2 (S)



1	Rua do beco	
2	Sem rua	
3	Subúrbio	

 $\Rightarrow$ 

1	Pedro

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 7

Correspondência encontrada Registro é retornado

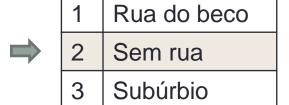
#### Tab1 (R)

4	Joao
7	Ana
3	Maria



1	Pedro

### Tab 2 (S)



4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 7

Tab2 avança

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



1	Pedro

### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



1	Pedro

### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 8

Tab2 avança

# **Nested-Loop Join**

#### Tab1 (R)

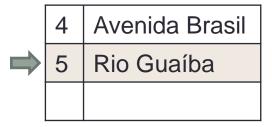
4	Joao
7	Ana
3	Maria



1	Pedro

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 8

# **Nested-Loop Join**

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



1	Pedro

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

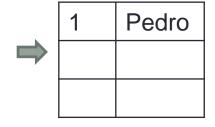
	4	Avenida Brasil
	5	Rio Guaíba
$\Rightarrow$		

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 8

# **Nested-Loop Join**

#### Tab1 (R)

4	Joao
7	Ana
3	Maria



### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 8

# BLOCK NESTED LOOP JOIN

 O algoritmo BlockNestedLoopJoin (BNLJ) processa um conjunto de registros de r em vez de um registro de cada vez

- Isso reduz o custo em transferências de blocos de
  - Br+ Nr \* Bs
- para
  - Br+ Br \* Bs

- O BNLJ pode ser implementado por meio de um buffer
- Processamento básico
  - registros da relação externa são carregados até encher o Join Buffer.
  - para cada registro da relação interna, o buffer é verificado
- Essa é a estratégia usada pelo MySQL para implementar o BNLJ

- O exemplo a seguir mostra como funcionaria a varredura das tabelas usando o BNLJ
- O algoritmo assume que um registro de um lado pode ter correspondências com mais do que um registro do outro lado
  - Ou seja, ele não para de procurar quando uma correspondência for encontrada
- O buffer tem espaço para três registros

Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer

Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 0Transf. blocos Tab 2 = 0

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

4	Joao
7	Ana
3	Maria

#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 0

O buffer é preenchido

#### Tab1 (R)

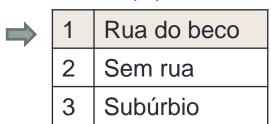
4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

4	Joao
7	Ana
3	Maria

#### Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

O cursor em tab2 é aberto e vai para o primeiro registro

Tab1 (R)

Joao 7 Ana 3 Maria

Pedro



Join buffer

4	Joao
7	Ana
3	Maria

Tab 2 (S)

$\Rightarrow$	1	Rua do beco
	2	Sem rua
	3	Subúrbio

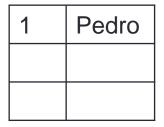
4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

O cursor do buffer vai para o primeiro registro

Tab1 (R)

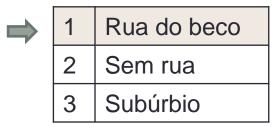
4	Joao
7	Ana
3	Maria



Join buffer



Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Tab1 (R)

4 Joao7 Ana3 Maria

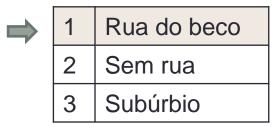


Join buffer

4	Joao
7	Ana
3	Maria

1	Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

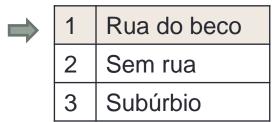
1	Pedro

Join buffer

4	Joao
7	Ana
3	Maria



Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

#### Tab1 (R)

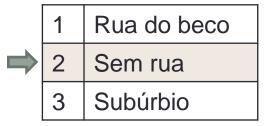
4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

4	Joao
7	Ana
3	Maria

#### Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

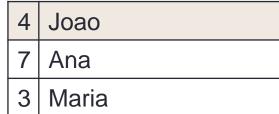
Tab1 (R)

4 Joao 7 Ana 3 Maria

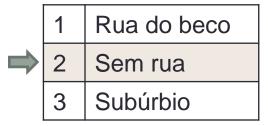
Pedro



Join buffer



Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

O cursor do buffer vai para o primeiro registro

Tab1 (R)

4 Joao7 Ana3 Maria

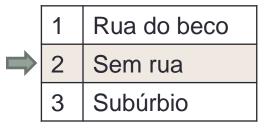


Join buffer



1	Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Tab1 (R)

4	Joao
7	Ana
3	Maria

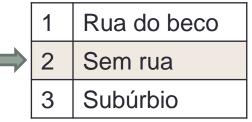


Join buffer

4	Joao
7	Ana
3	Maria

1 Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

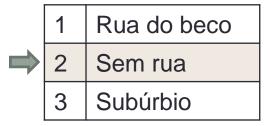
1	Pedro

Join buffer

4	Joao
7	Ana
3	Maria



Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

4	Joao
7	Ana
3	Maria

### Tab 2 (S)

	1	Rua do beco
	2	Sem rua
$\Rightarrow$	3	Subúrbio

4	Avenida Brasil
5	Rio Guaíba

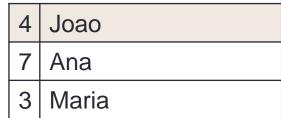
Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Tab1 (R)

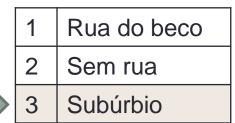
4 Joao7 Ana3 Maria



Join buffer



Tab 2 (S)



1 Pedro

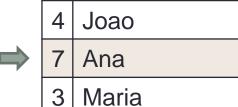
4 Avenida Brasil5 Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

O cursor do buffer vai para o primeiro registro

Tab1 (R)

Joao 4 7 Ana 3 Maria

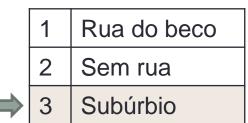


Pedro

Join buffer



Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Tab1 (R)



 $\Rightarrow$ 

Join buffer

4	Joao
7	Ana
3	Maria

1 Pedro

Tab 2 (S)

	1	Rua do beco
	2	Sem rua
$\Rightarrow$	3	Subúrbio

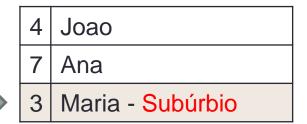
4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Tab1 (R)

4 Joao7 Ana3 Maria

Join buffer

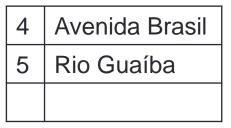


1 Pedro

3, Maria - Subúrbio

Tab 2 (S)

	1	Rua do beco
	2	Sem rua
$\Rightarrow$	3	Subúrbio



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Correspondência encontrada!

O registro é retornado.

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

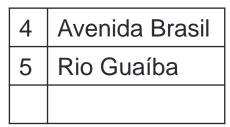
Join buffer

4	Joao
7	Ana
3	Maria - Subúrbio



Tab 2 (S)

	1	Rua do beco
	2	Sem rua
$\Rightarrow$	3	Subúrbio



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 1

Buffer acabou

### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro	

#### Join buffer

4	Joao
7	Ana
3	Maria - Subúrbio

### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Tab2 avança

Tab1 (R)

4	Joao
7	Ana
3	Maria

Pedro



Join buffer

4	Joao
7	Ana
3	Maria - Subúrbio

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

O cursor do buffer vai para o primeiro registro

Tab1 (R)

4	Joao
7	Ana
3	Maria

Join buffer



1 Pedro

4, Joao – Avenida Brasil

Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

 $\Rightarrow$ 

4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Correspondência encontrada!

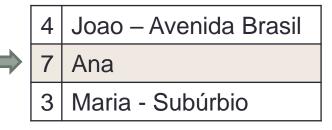
O registro é retornado.

Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer



1	Pedro

Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Tab1 (R)

4	Joao
7	Ana
3	Maria



- 1										
- 1		ı		١.	h		ιt	+		r
J	O	ı			U	ľ	ш	н	ㄷ	ш
_	_	-	-	-		_		-	_	-

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio

1	Pedro



1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio



### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1

Transf. blocos Tab 2 = 2

Buffer acabou

### Tab1 (R)

4	Joao
7	Ana
3	Maria

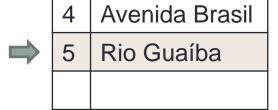
1	Pedro

#### Join buffer

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio

#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Tab2 avança

Tab1 (R)

4	Joao
7	Ana
3	Maria



Join buffer

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio

1	Pedro

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

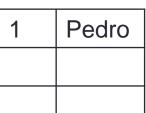
Avenida Brasil Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

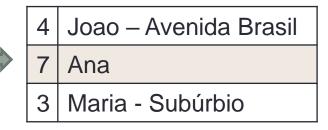
O cursor do buffer vai para o primeiro registro

Tab1 (R)

4	Joao
7	Ana
3	Maria

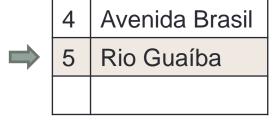


Join buffer



Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Tab1 (R)

4	Joao
7	Ana
3	Maria



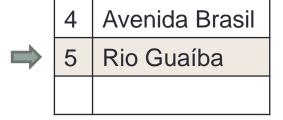
Join buffer

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio

1	Pedro

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

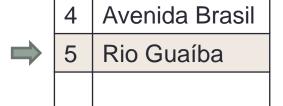
#### Join buffer

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio



#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio



Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Buffer acabou

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

4	Joao – Avenida Brasil
7	Ana
3	Maria - Subúrbio

#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

	4	Avenida Brasil
	5	Rio Guaíba
<b>⇒</b>		

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

Acabaram os registros em tab2

Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer

Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 1Transf. blocos Tab 2 = 2

O buffer é esvaziado

Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer

1	Pedro

Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 2

Mais registros são carregados

Tab1 (R)

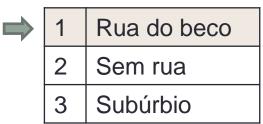
4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer

1	Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

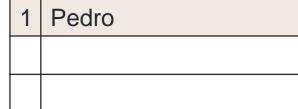
O cursor em tab2 volta para o primeiro registro

Tab1 (R)

4 Joao7 Ana3 Maria

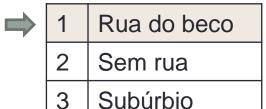


- 1				- 1			•	e		
- 1	O	п			h	1.1	•	+		10
	( )		11		r 1				_	
v	v				$\sim$	u			v	



1	Pedro





4	Avenida Brasil
5	Rio Guaíba

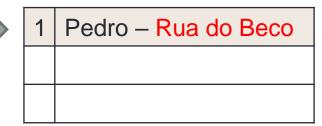
Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

O cursor do buffer vai para o primeiro registro

Tab1 (R)

4	Joao
7	Ana
3	Maria

Join buffer



1 Pedro

1, Pedro – Rua do Beco

Tab 2 (S)



1	Rua do beco
2	Sem rua
3	Subúrbio

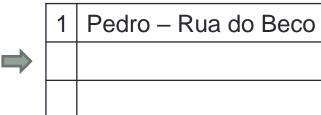
4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

Correspondência encontrada Registro é retornado

Tab1 (R)

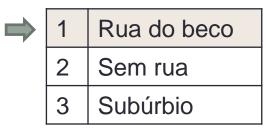




Join buffer

1	Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

Buffer acabou

#### Tab1 (R)

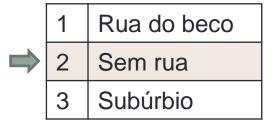
4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

1	Pedro – Rua do Beco

#### Tab 2 (S)

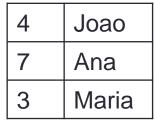


4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

Tab2 avança

Tab1 (R)

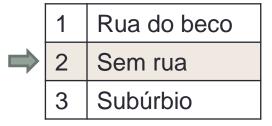


Join buffer

1	Pedro – Rua do Beco

Pedro

Tab 2 (S)



4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

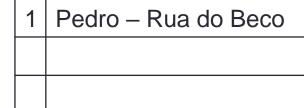
O cursor do buffer é posicionado no primeiro registro

Tab1 (R)





Join buffer



1	Pedro

Tab 2 (S)

	1	Rua do beco
$\Rightarrow$	2	Sem rua
	3	Subúrbio

4	Avenida Brasil
5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

Buffer acabou

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

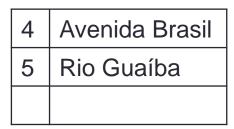
1	Pedro

#### Join buffer

1	Pedro – Rua do Beco

#### Tab 2 (S)

	1	Rua do beco	
	2	Sem rua	
>	3	Subúrbio	



Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

Tab2 avança

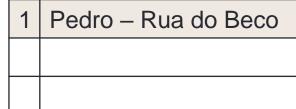
Tab1 (R)

Joao 7 Ana 3 Maria

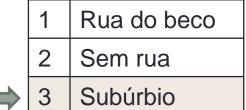
Pedro



Join buffer



Tab 2 (S)



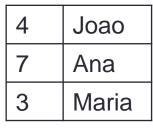
4	Avenida Brasil	
5	Rio Guaíba	

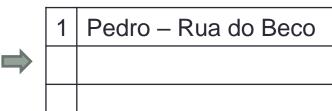
Transf. blocos Tab 1 = 2

Transf. blocos Tab 2 = 3

O cursor do buffer é posicionado no primeiro registro

Tab1 (R)



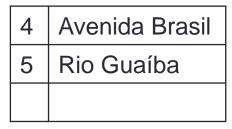


Join buffer

Pedro

Tab 2 (S)

	1	Rua do beco
	2	Sem rua
$\Rightarrow$	3	Subúrbio



Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 3

Buffer acabou

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

1	Pedro – Rua do Beco

#### Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

$\Rightarrow$	4	Avenida Brasil
	5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

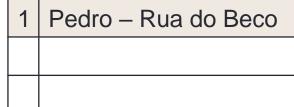
Tab2 avança

Tab1 (R)

Joao Ana 3 Maria



Join buffer



1	Pedro

1	Pedro

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil
5	Rio Guaíba

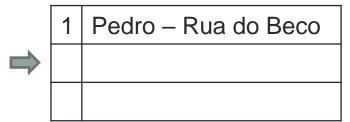
Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

O cursor do buffer é posicionado no primeiro registro

Tab1 (R)



Join buffer



1	Pedro

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

$\Rightarrow$	4	Avenida Brasil
	5	Rio Guaíba

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

Buffer acabou

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

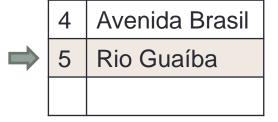
1	Pedro

#### Join buffer

1	Pedro – Rua do Beco	

#### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

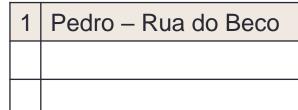
Tab2 avança

Tab1 (R)

4 Joao7 Ana3 Maria



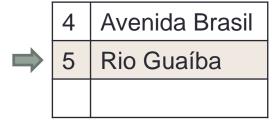
Join buffer



1	Pedro

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	



Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

O cursor do buffer é posicionado no primeiro registro

Tab1 (R)

4 Joao7 Ana3 Maria



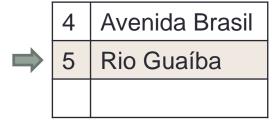
Join buffer

1 Pedro – Rua do Beco

1	Pedro

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	3 Subúrbio	



Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

Buffer acabou

#### Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

#### Join buffer

1	Pedro – Rua do Beco

#### Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

	4	Avenida Brasil
	5	Rio Guaíba
$\Rightarrow$		

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

Tab2 acabou

Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer

Tab 2 (S)

1	Rua do beco	
2	Sem rua	
3	Subúrbio	

4	Avenida Brasil	
5	Rio Guaíba	

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

O buffer é esvaziado

Tab1 (R)

4	Joao
7	Ana
3	Maria

1	Pedro

Join buffer

Tab 2 (S)

1	Rua do beco
2	Sem rua
3	Subúrbio

4	Avenida Brasil		
5	Rio Guaíba		

Transf. blocos Tab 1 = 2Transf. blocos Tab 2 = 4

Não há mais registros de tab1 para processar

# Comparação

- Custo NLJ
  - Br+ Nr \* Bs
- Custo BNLJ
  - Br+ Br \* Bs

Números

• 
$$Br = 2$$

• 
$$Nr = 4$$

• 
$$Bs = 2$$

Tabela	NLJ	BNLJ
Tab1	2	2
tab2	8	4

Blocos transferidos

# OBJETIVO DO TRABALHO

## Objetivo do Trabalho

- O objetivo do trabalho é implementar o BlockNestedLoopJoin
  - Criar uma classe chamada xxxBlockNestedLoopJoin, onde xxx é o nome do aluno
  - Implementar as funções herdadas das interfaces

```
public void open() throws Exception;
public Tuple next() throws Exception;
public boolean hasNext() throws Exception;
public void close();
public Operation getLeftOperation();
public Operation getRigthOperation();
```

## Resumo das principais funções

- Função open()
  - Prepara tudo para que se comece a fazer a junção (abre cursores, limpa variáveis, ...)
- Funções next()
  - Recupera o próximo registro

- Funções hasNext()
  - Verifica se há mais elementos a recuperar

## Observações

- É preciso cuidado na implementação das funções hasNext() e next().
  - Por exemplo, deve-se garantir que o iterador só avance quando for chamado o next().
- Analise as operações já criadas para auxiliar na criação do algoritmo
  - Principalmente o NestedLoopJoin

Use a função createTable para testar as operações

```
Table table = createTable("c:\\teste\\ibd","t1.ibd",1000, true, 1);

Operation s1 = new TableScan(table1);
...
```

#### Parâmetros:

- Pasta onde está o banco
- Arquivo onde está o banco
- Quantidade de registros
- Flag indicando se registros ficarão desordenados
- Distância entre um valor e outro

```
Table table1 = createTable("c:\\teste\\ibd","t1.ibd",50, false, 1);
Table table2 = createTable("c:\\teste\\ibd","t2.ibd",100, false, 1);
Operation join = new NestedLoopJoin(scan1, scan2);
QueryOptimizer opt = new QueryOptimizer();
Operation query = opt.optimizeQuery(join);
Params.BLOCKS LOADED = 0;
Params.BLOCKS\_SAVED = 0;
query.open();
while (query.hasNext()){
    Tuple r = query.next();
    System.out.println(r.primaryKey + " - "+r.content);
System.out.println("blocks loaded " + Params.BLOCKS_LOADED);
System.out.println("blocks saved " + Params.BLOCKS SAVED);
```

#### Criação das tabelas

```
Table table1 = createTable("c:\\teste\\ibd","t1.ibd",50, false, 1);
Table table2 = createTable("c:\\teste\\ibd","t2.ibd",100, false, 1);
Operation join = new NestedLoopJoin(scan1, scan2);
QueryOptimizer opt = new QueryOptimizer();
Operation query = opt.optimizeQuery(join);
Params.BLOCKS LOADED = 0;
Params.BLOCKS\_SAVED = 0;
query.open();
while (query.hasNext()){
    Tuple r = query.next();
    System.out.println(r.primaryKey + " - "+r.content);
System.out.println("blocks loaded " + Params.BLOCKS_LOADED);
System.out.println("blocks saved " + Params.BLOCKS SAVED);
```

#### Criação da consulta

```
Table table1 = createTable("c:\\teste\\ibd","t1.ibd",50, false, 1);
Table table2 = createTable("c:\\teste\\ibd","t2.ibd",100, false, 1);
Operation join = new NestedLoopJoin(scan1, scan2);
QueryOptimizer opt = new QueryOptimizer();
Operation query = opt.optimizeQuery(join);
Params.BLOCKS LOADED = 0;
Params.BLOCKS\_SAVED = 0;
query.open();
while (query.hasNext()){
    Tuple r = query.next();
    System.out.println(r.primaryKey + " - "+r.content);
System.out.println("blocks loaded " + Params.BLOCKS_LOADED);
System.out.println("blocks saved " + Params.BLOCKS_SAVED);
```

#### Otimização da consulta

```
Table table1 = createTable("c:\\teste\\ibd","t1.ibd",50, false, 1);
Table table2 = createTable("c:\\teste\\ibd","t2.ibd",100, false, 1);
Operation join = new NestedLoopJoin(scan1, scan2);
QueryOptimizer opt = new QueryOptimizer();
Operation query = opt.optimizeQuery(join);
Params.BLOCKS LOADED = 0;
Params.BLOCKS\_SAVED = 0;
query.open();
while (query.hasNext()){
    Tuple r = query.next();
    System.out.println(r.primaryKey + " - "+r.content);
System.out.println("blocks loaded " + Params.BLOCKS_LOADED);
System.out.println("blocks saved " + Params.BLOCKS_SAVED);
```

#### Execução da consulta

```
Table table1 = createTable("c:\\teste\\ibd","t1.ibd",50, false, 1);
Table table2 = createTable("c:\\teste\\ibd","t2.ibd",100, false, 1);
Operation join = new NestedLoopJoin(scan1, scan2);
QueryOptimizer opt = new QueryOptimizer();
Operation query = opt.optimizeQuery(join);
Params.BLOCKS LOADED = 0;
Params.BLOCKS\_SAVED = 0;
query.open();
while (query.hasNext()){
    Tuple r = query.next();
    System.out.println(r.primaryKey + " - "+r.content);
System.out.println("blocks loaded " + Params.BLOCKS_LOADED);
System.out.println("blocks saved " + Params.BLOCKS SAVED);
```

#### Medição de desempenho

```
Table table1 = createTable("c:\\teste\\ibd","t1.ibd",50, false, 1);
Table table2 = createTable("c:\\teste\\ibd","t2.ibd",100, false, 1);
Operation join = new NestedLoopJoin(scan1, scan2);
QueryOptimizer opt = new QueryOptimizer();
Operation query = opt.optimizeQuery(join);
Params.BLOCKS LOADED = 0;
Params.BLOCKS_SAVED = 0;
query.open();
while (query.hasNext()){
    Tuple r = query.next();
    System.out.println(r.primaryKey + " - "+r.content);
System.out.println("blocks loaded " + Params.BLOCKS_LOADED);
System.out.println("blocks saved " + Params.BLOCKS SAVED);
```

- A classe QueryOptimizer usa o algoritmo NestedLoopJoin
  - Troque pelo BlockNestedLoopJoin
- Verificação de consistência
  - Analise se os registros retornados pela consulta com NLJ e a consulta com BNLJ são os mesmos
- Verificação de desempenho
  - Analise se o número de blocos carregados usando BNLJ é menor do que NLJ

- O número de transferências de bloco deve ser proporcional à seguinte equação
  - Br + Br' \* Bs
- Onde
  - Br = número total de blocos de R (63)
  - Br' = número de blocos de R que possuem registros
  - Bs = número total de blocos de S (63)

- Mantenha o tamanho do buffer de blocos com o tamanho original em BufferManager
- O join buffer dentro do BlockNestedLoopJoin deve ter o tamanho equivalente a Block.RECORDS\_AMOUNT

- Teste com diferentes consultas, variando
  - quantidade de tabelas
  - quantidade de registros de cada tabela
  - ordenação de cada tabela

#### Obs1.

 A consulta original pode usar somente TableScans e NestedLoopJoins

#### Obs2

 Caso os dados estejam desordenados, a ordem dos registros no resultado pode variar entre o NLJ e o BNLJ

# Entrega

- Prazo final de entrega, sem descontos
  - Segunda, 25 de maio às 23:55
- A cada dia de atraso, a nota é decrementada em 50%.

- O que entregar
  - O código fonte da classe de otimização (.java) não comprimido