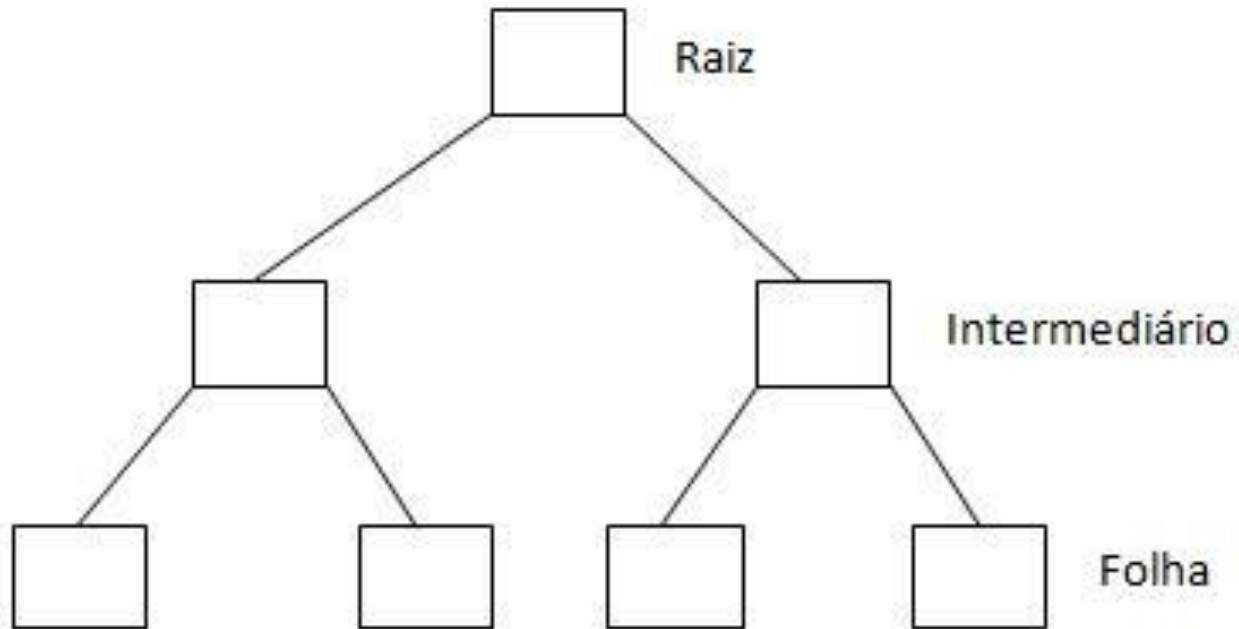
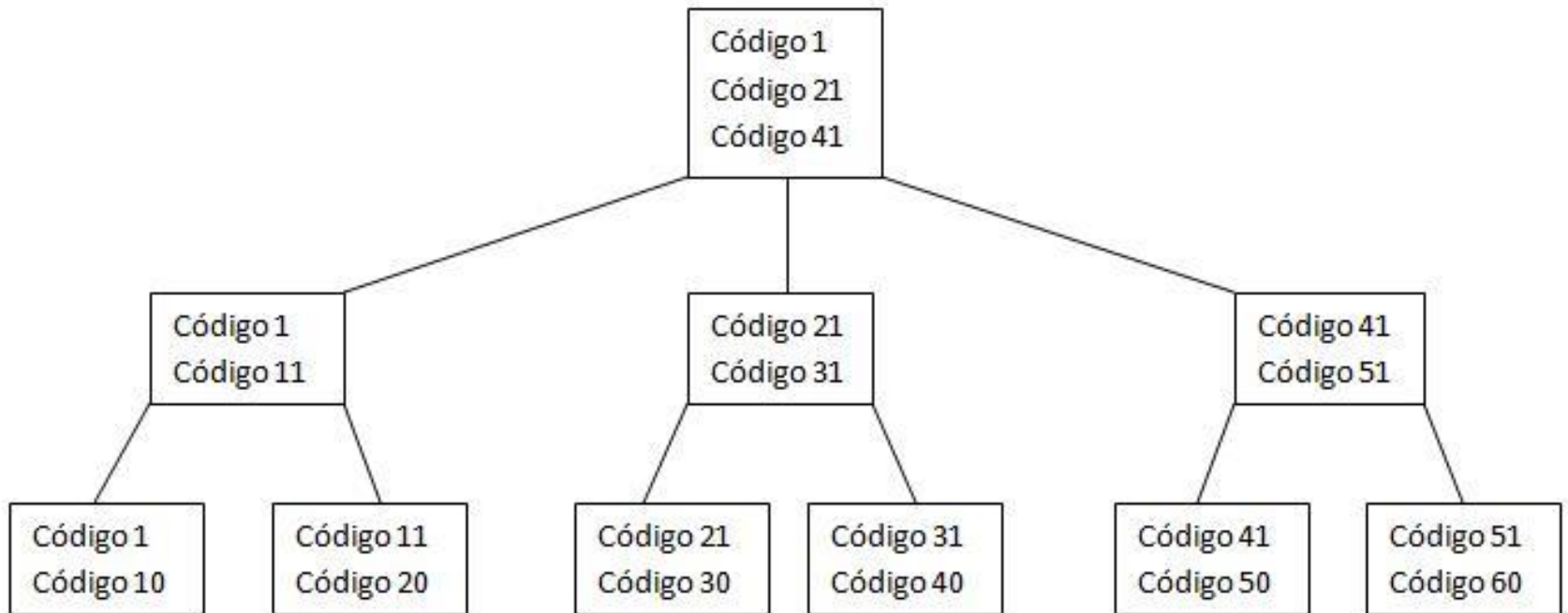


# INDEX

# B-TREE



# B-TREE



# B-TREE

A maneira mais básica de localizar um registro é procurando-o, na sequência de registros, até encontrá-lo. É o que chamamos tecnicamente de "*table scan*", ou seja, varrer a tabela toda...

Este método pode ser eficiente quando o registro buscado for o primeiro da lista. Mas e se for o último?!

Um jeito interessante de resolver este problema é ordenar a tabela pelo campo em questão.

Neste caso, pode ser realizada uma busca por métodos de "biseção", ou seja, dividir a tabela ao meio e avaliar em que metade o registro se encontra.

Para isso, compara-se o valor do registro na posição equivalente ao "meio" da tabela com o valor buscado. Se for menor, o dado que procuramos está na metade superior. Se for maior, está na metade inferior.

Depois, repetimos o mesmo processo para a metade que sobrou.

# B-TREE

Vejam como funciona na prática: vamos procurar, numa lista de 54 nomes femininos de origem Tupi, o nome **PAQUETÁ**.

A metade na qual o nome se encontra foi copiada à direita da lista anterior, e assim sucessivamente até encontrarmos o nome **PAQUETÁ**. Repare que foram necessários somente **6 passos**, muito menos que os **45** que seriam necessários no caso de “*table scan*”.

# B-TREE

ABANA	CECI	MARANEÍMA	MARANEÍMA	MARANEÍMA	NAARA	PAQUETÁ
ACI	COARACI	MARICÍ / MARICY	MARICÍ / MARICY	MARICÍ / MARICY	PAQUETÁ	
AÇUCENA	EÇAIARA	NAARA	NAARA	NAARA		
AIMORÉ	GUACIRA	PAQUETÁ	PAQUETÁ	PAQUETÁ		
AIRUMÃ	GUAYI	PAVUNA	PAVUNA			
AIRY	HURASSÍ	PIATÃ	PIATÃ			
AISÓ	IACINA	QUARACIEMA	QUARACIEMA			
AJURICABA	IBÁ	SABARÁ				
AJYRA	IPUÃ	SEPETIBA				
AMARY	IRACI	TACIQUARA				
ANAJÁ	JACICOÊ	TAÍÁ				
ANAMI	JANDAIA	UAIANA				
ANECI	LACIRANDY	UAUIARARÁ				
ARACÊ	MANACÁ					
ARACEMA	MARANEÍMA					
ARACI	MARICÍ / MARICY					
ARAÍBA	NAARA					
ARANI	PAQUETÁ					
ARAPUÃ	PAVUNA					
ARARIBÓIA	PIATÃ					
BARAÚNA	QUARACIEMA					
ÇAÇAPAVA	SABARÁ					
CAETÉ	SEPETIBA					
CAMAPUÃ	TACIQUARA					
CÂNDIA	TAÍÁ					
CATIRA	UAIANA					
CAYRES	UAUIARARÁ					
CECI						
COARACI						
EÇAIARA						
GUACIRA						
GUAYI						
HURASSÍ						
IACINA						
IBÁ						
IPUÃ						
IRACI						
JACICOÊ						
JANDAIA						
LACIRANDY						
MANACÁ						
MARANEÍMA						
MARICÍ / MARICY						
NAARA						
PAQUETÁ						
PAVUNA						
PIATÃ						
QUARACIEMA						
SABARÁ						
SEPETIBA						
TACIQUARA						
TAÍÁ						
UAIANA						
UAUIARARÁ						

# Sintaxe

- CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX *index\_name*
- ON( *column* [ ASC | DESC ] [ ,...*n* ] )
- [ INCLUDE ( *column\_name* [ ,...*n* ] ) ]
- [ WITH ( [ ,...*n* ] ) ]
- [ ON { *partition\_scheme\_name* ( *column\_name* )
- | *filegroup\_name*
- | default
- }

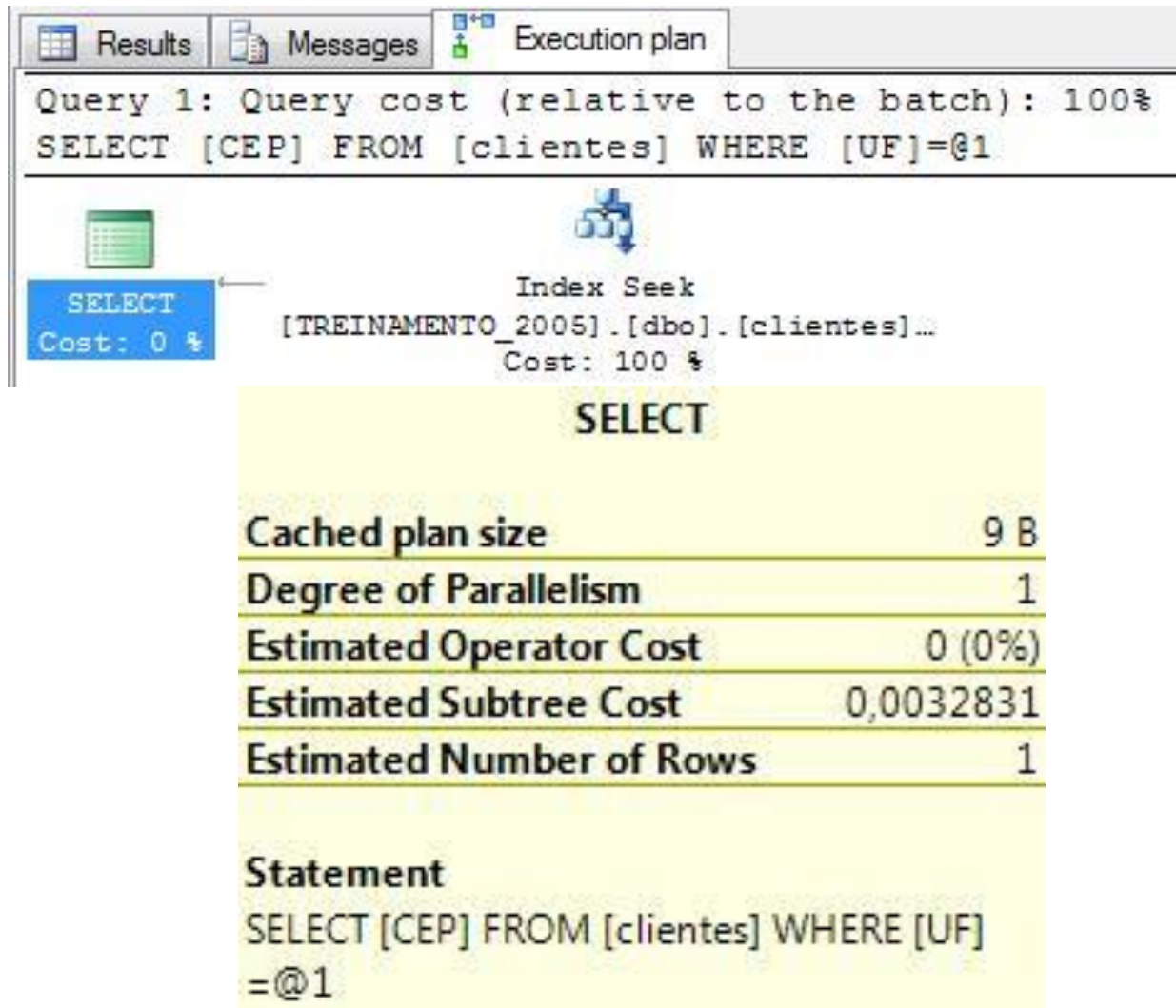
# Métodos de Acesso

Os acessos aos dados das tabelas e índices podem ser de duas formas, SEEK ou SCAN.

- **SCAN** - busca em TODOS os elementos da estrutura (que pode ser uma tabela ou um índice). É usado quando não possui índices que atendam a instrução de select ou quando a quantidade de registros que a query retorna (em percentual) é grande.
- **SEEK** - busca binária nos elementos de um índice. É usado quando existe um índice que é adequado e a quantidade de registros (em percentual) retornados é pequena.
- Sendo assim, é possível executar as seguintes operações para acesso nas tabelas/índices:
- **TABLE SCAN** - Busca em todos os elementos da tabela, de forma seqüencial;
- **INDEX SCAN** - Busca em todos os elementos de um índice nonclustered, de forma seqüencial;
- **INDEX SEEK** - Busca binária num índice nonclustered;
- **CLUSTERED INDEX SCAN** - Busca em todos os elementos de um índice clustered, de forma seqüencial;
- **CLUSTERED INDEX SEEK** - Busca binária num índice clustered.



# Desempenho do B-TREE



Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT [CEP] FROM [clientes] WHERE [UF]=@1
```

SELECT  
Cost: 0 %

Index Seek  
[TREINAMENTO\_2005].[dbo].[clientes]...  
Cost: 100 %

SELECT	
Cached plan size	9 B
Degree of Parallelism	1
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0,0032831
Estimated Number of Rows	1

**Statement**  
SELECT [CEP] FROM [clientes] WHERE [UF]  
=@1

# Exemplo

```
create database doismilhoes  
GO  
use doismilhoes
```

```
create table agenda(  
nome_completo varchar(100),  
endereco varchar(100))  
GO
```

```
create table joinagenda(  
codjoinagenda int identity not null,  
nome varchar(100))
```

# Exemplo

```
declare @cont as int
set @cont = 1
while @cont <=2000000
begin
    insert into agenda values ('Pessoa '+convert(varchar(10),@cont),'Rua
'+convert(varchar(10),@cont))
    insert into joinagenda values ('Pessoa '+convert(varchar(10),@cont))
    set @cont = @cont + 1
end
```

```
select * from agenda
```

```
select * from joinagenda
```

# Exemplo

```
SELECT * FROM sys.sysindexes  
WHERE id = OBJECT_ID('agenda')
```

```
SELECT * FROM sys.sysindexes  
WHERE id = OBJECT_ID('agenda')
```

```
CREATE NONCLUSTERED INDEX IX_agenda  
ON agenda (nome_completo)
```

```
CREATE NONCLUSTERED INDEX IX_joinagenda  
ON joinagenda (nome)
```

```
CREATE CLUSTERED INDEX IX_agenda  
ON agenda (nome_completo)
```

```
CREATE CLUSTERED INDEX IX_joinagenda  
ON joinagenda (nome)
```

```
DROP INDEX IX_agenda  
ON agenda
```

```
DROP INDEX IX_joinagenda  
ON joinagenda
```

# Exemplo

```
select endereco
from agenda
where nome_completo = 'Pessoa 1557854'
```

```
select agenda.nome_completo, agenda.endereco,
joinagenda.nome
from agenda
inner join joinagenda
on agenda.nome_completo = joinagenda.nome
where joinagenda.nome = 'Pessoa 1874362'
```

```
INSERT INTO joinagenda
VALUES ('Pessoa 1984784a')
```

```
DELETE joinagenda
WHERE nome = 'Pessoa 1984784a'
```

# Exemplo

- Sem INDEX

```
select agenda.nome_completo, agenda.endereco, joinagenda.nome
from agenda
inner join joinagenda
on agenda.nome_completo = joinagenda.nome
where joinagenda.nome = 'Pessoa 1874362'
```

Mensagens Plano de execução

Consulta 1: Custo da consulta (relativo ao lote): 100%

select agenda.nome\_completo, agenda.endereco, joinagenda.nome from agenda inner join joinagenda on agenda.nome\_completo...

Índice Ausente (Impacto 60.247): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[agenda] ([nome\_...

```
graph TD
    SELECT[SELECT  
Custo: 0 %] --> NL[Nested Loops  
(Inner Join)  
Custo: 5 %]
    NL --> TS1[Table Scan  
[agenda]  
Custo: 55 %]
    NL --> P[Parallelism  
(Gather Streams)  
Custo: 2 %]
    P --> TS2[Table Scan  
[joinagenda]  
Custo: 38 %]
```

Consulta executada com êxito. | (local) (10.0 SP3) | sa (53) | doismilhoes | 00:00:00 | 0 linhas

# Exemplo

- Com INDEX NONCLUSTERED em ambas

```
select agenda.nome_completo, agenda.endereco, joinagenda.nome
from agenda
inner join joinagenda
on agenda.nome_completo = joinagenda.nome
where joinagenda.nome = 'Pessoa 1874362'
```

Mensagens Plano de execução

Consulta 1: Custo da consulta (relativo ao lote): 100%

select agenda.nome\_completo, agenda.endereco, joinagenda.nome from agenda inner join joinagenda on agenda.nome\_completo...

```
graph TD
    SELECT[SELECT  
Custo: 0 %] --> NL1[Nested Loops  
(Inner Join)  
Custo: 0 %]
    NL1 --> IS1[Index Seek (NonClustered)  
[joinagenda].[IX_joinagenda]  
Custo: 33 %]
    NL1 --> NL2[Nested Loops  
(Inner Join)  
Custo: 0 %]
    NL2 --> IS2[Index Seek (NonClustered)  
[agenda].[IX_agenda]  
Custo: 33 %]
    IS2 --> RID[RID Lookup (Heap)  
[agenda]  
Custo: 33 %]
```

Consulta executada com êxito. (local) (10.0 SP3) sa (53) doismilhoes 00:00:00 0 linhas

# Exemplo

- Com INDEX NONCLUSTERED em ambas

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the following SQL script:

```
INSERT INTO joinagenda
VALUES ('Pessoa 1984784a')

DELETE joinagenda
WHERE nome = 'Pessoa 1984784a'
```

The bottom pane shows the execution plan for two queries:

**Consulta 1: Custo da consulta (relativo ao lote): 46%**  
INSERT INTO joinagenda VALUES ('Pessoa 1984784a')

The execution plan for the INSERT query consists of the following operators (from right to left):

- Constant Scan (Custo: 0 %)
- Compute Scalar (Custo: 0 %)
- Compute Scalar (Custo: 0 %)
- Table Insert (Custo: 100 %)
- INSERT (Custo: 0 %)

**Consulta 2: Custo da consulta (relativo ao lote): 54%**  
DELETE joinagenda WHERE nome = 'Pessoa 1984784a'

The execution plan for the DELETE query consists of the following operators (from right to left):

- Index Seek (NonClustered) [joinagenda].[IX\_joinagenda] (Custo: 14 %)
- Top (Custo: 0 %)
- Table Delete (Custo: 86 %)
- DELETE (Custo: 0 %)

The status bar at the bottom indicates: **Consulta executada com êxito.** (local) (10.0 SP3) sa (53) doismilhoes 00:00:00 0 linhas



# Exemplo

- Com INDEX CLUSTERED em ambas

```
select agenda.nome_completo, agenda.endereco, joinagenda.nome
from agenda
inner join joinagenda
on agenda.nome_completo = joinagenda.nome
where joinagenda.nome = 'Pessoa 1874362'
```

Mensagens Plano de execução

Consulta 1: Custo da consulta (relativo ao lote): 100%

select agenda.nome\_completo, agenda.endereco, joinagenda.nome from agenda inner join joinagenda on agenda.nome\_completo...

```
graph TD
    SELECT[SELECT  
Custo: 0 %] --- NL[Nested Loops  
(Inner Join)  
Custo: 0 %]
    NL --- CS1[Clustered Index Seek (Clustered)  
[joinagenda].[IX_joinagenda]  
Custo: 50 %]
    NL --- CS2[Clustered Index Seek (Clustered)  
[agenda].[IX_agenda]  
Custo: 50 %]
```

Consulta executada com êxito.

(local) (10.0 SP3) sa (53) doismilhoes 00:00:00 0 linhas

# Exemplo

- Com INDEX CLUSTERED em ambas

INSERT INTO joinagenda  
VALUES ('Pessoa 1984784a')

DELETE joinagenda  
WHERE nome = 'Pessoa 1984784a'

Mensagens Plano de execução

Consulta 1: Custo da consulta (relativo ao lote): 43%  
INSERT INTO joinagenda VALUES ('Pessoa 1984784a')

INSERT  
Custo: 0 %

Clustered Index Insert  
[joinagenda].[IX\_joinagenda]  
Custo: 100 %

Compute Scalar  
Custo: 0 %

Compute Scalar  
Custo: 0 %

Constant Scan  
Custo: 0 %

Consulta 2: Custo da consulta (relativo ao lote): 57%  
DELETE joinagenda WHERE nome = 'Pessoa 1984784a'

DELETE  
Custo: 0 %

Clustered Index Delete  
[joinagenda].[IX\_joinagenda]  
Custo: 100 %

Consulta executada com êxito. (local) (10.0 SP3) sa (53) doismilhoes 00:00:00 0 linhas