SQLServerFast.com
Execution Plan Video Training

Block 2: Reading data

Level: Basic

Chapter 2: Scan operators

Scan operators

Eight different scan operators

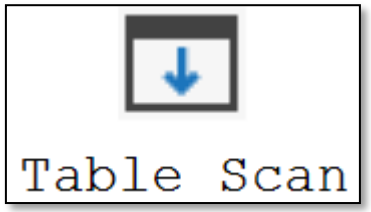
- Clustered Index Scan
- Columnstore Index Scan
- Constant Scan
- Deleted Scan
- Index Scan
- Inserted Scan
- Remote Scan
- Table Scan

Scan operators

- Eight different scan operators
- Read all data from their source
- Scan vs seek?
  - Seek operators read only a specific subset of the data
    - For small subset, seek is faster than scan
      - Cost of enabling the seek might exceed savings
  - Scan operators read from start to finish
    - Or until parent operator stops asking for data
      - May be limited by Top, or by other operators

Scan operators

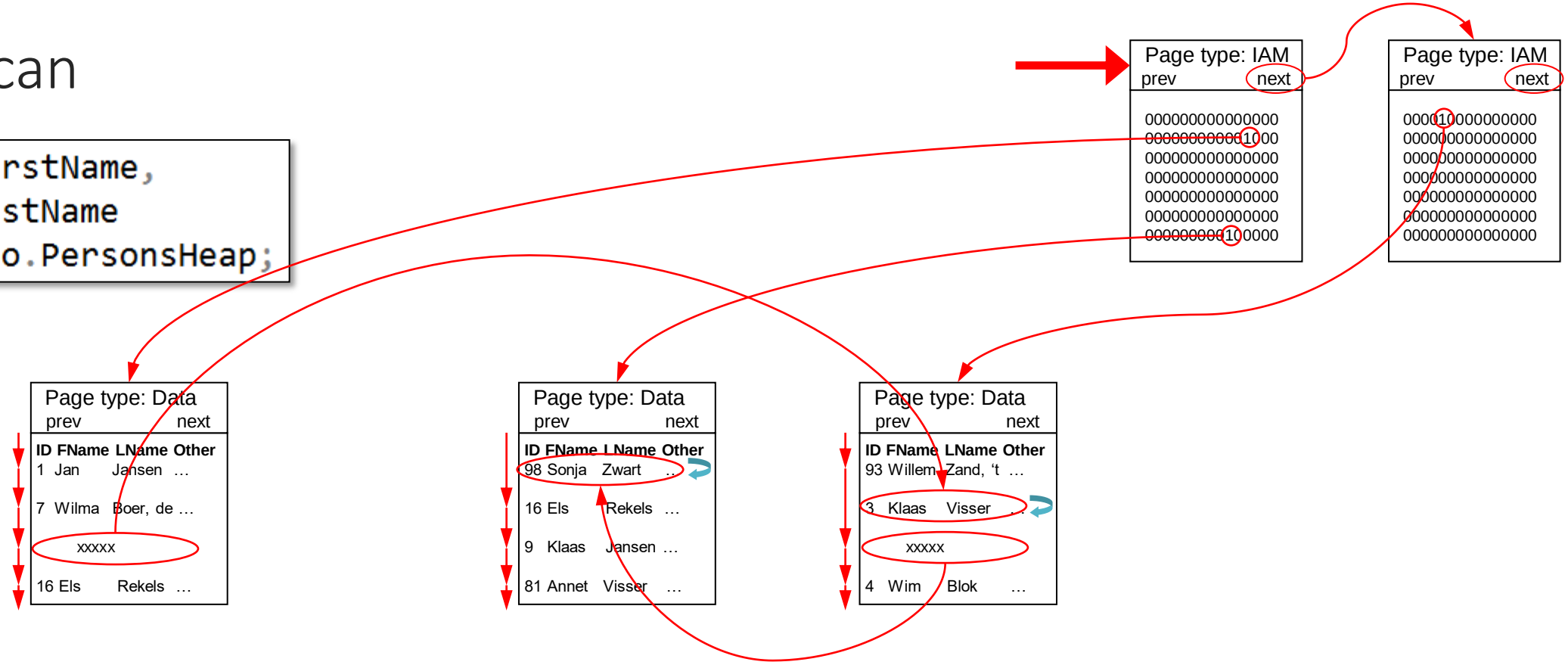
- Table Scan
  - Reads (all) data from a heap table
    - Heap table: Table without clustered index
      - Should (generally) be avoided
      - Choose a clustered index for all your tables



Scan operators

Table Scan

```
SELECT FirstName,
       LastName
FROM   dbo.PersonsHeap;
```



Scan operators

Table Scan

Properties (popup)

Actual and estimated row count

“Pushed down” predicate

Which table is scanned?

Data returned to caller

Table Scan

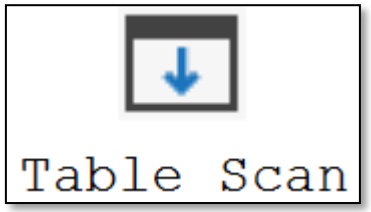
Scan rows from a table.

Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	1596
Actual Number of Rows for All Executions	1596
Actual Number of Batches	0
Estimated I/O Cost	0,581644
Estimated Operator Cost	0,583556 (100%)
Estimated CPU Cost	0,0019126
Estimated Subtree Cost	0,583556
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1596
Estimated Number of Rows to be Read	1596
Estimated Row Size	285 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	0

Predicate
[AdventureWorks2017].[dbo].[DatabaseLog].[DatabaseUser] as [dl].[DatabaseUser]=N'dbo'

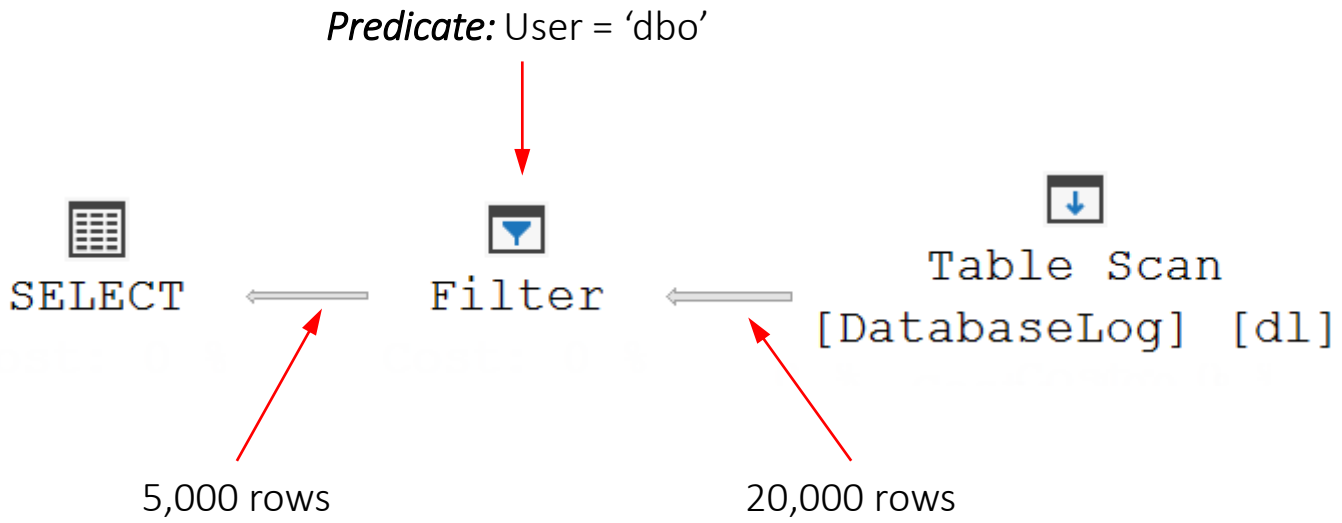
Object
[AdventureWorks2017].[dbo].[DatabaseLog] [dl]

Output List
[AdventureWorks2017].[dbo].[DatabaseLog].PostTime;
[AdventureWorks2017].[dbo].[DatabaseLog].Event;
[AdventureWorks2017].[dbo].[DatabaseLog].Object



Scan operators

Predicate pushdown



Scan operators

Table Scan

Properties (popup)

Number of rows the operator read
(includes rows read but not returned)

"Pushed down" predicate

Table Scan

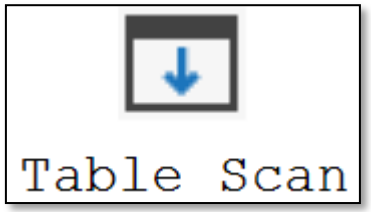
Scan rows from a table.

Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	1596
Actual Number of Rows for All Executions	1596
Actual Number of Batches	0
Estimated I/O Cost	0,581644
Estimated Operator Cost	0,583556 (100%)
Estimated CPU Cost	0,0019126
Estimated Subtree Cost	0,583556
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1596
Estimated Number of Rows to be Read	1596
Estimated Row Size	285 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	0

Predicate
[AdventureWorks2017].[dbo].[DatabaseLog].[DatabaseUser] as [dl].[DatabaseUser]=N'dbo'

Object
[AdventureWorks2017].[dbo].[DatabaseLog] [dl]

Output List
[AdventureWorks2017].[dbo].[DatabaseLog].PostTime;
[AdventureWorks2017].[dbo].[DatabaseLog].Event;
[AdventureWorks2017].[dbo].[DatabaseLog].Object



Scan operators

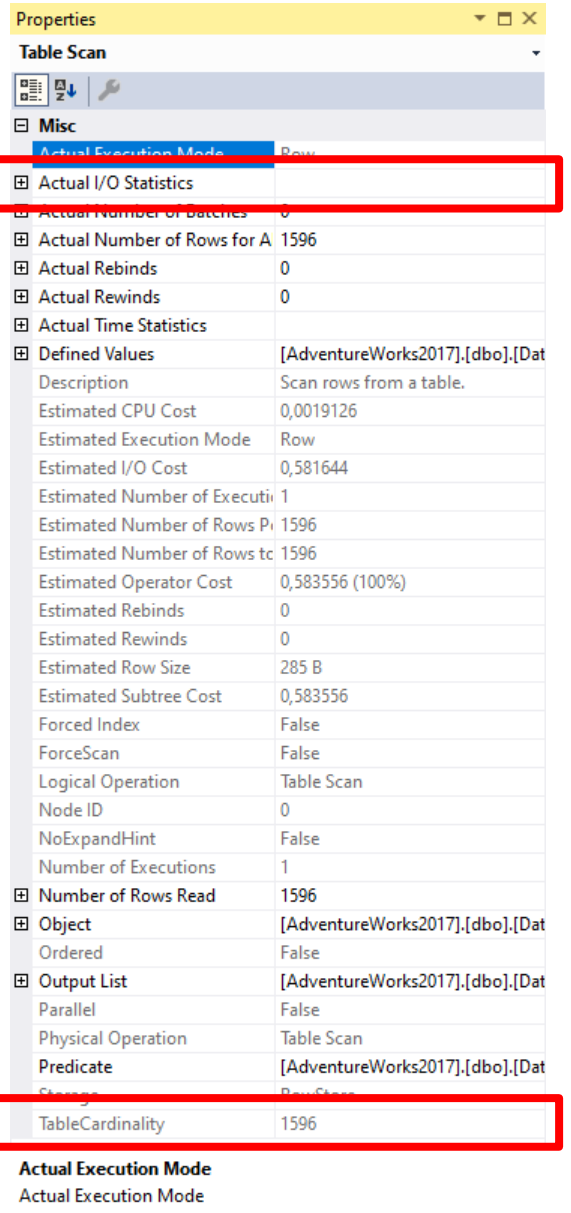
Table Scan

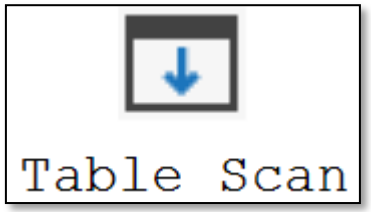
Properties (full list)

How many reads?

Actual I/O Statistics	
Actual Lob Logical Reads	0
Actual Lob Physical Reads	0
Actual Lob Read Aheads	0
Actual Logical Reads	782
Actual Physical Reads	0
Actual Read Aheads	0
Actual Scans	1

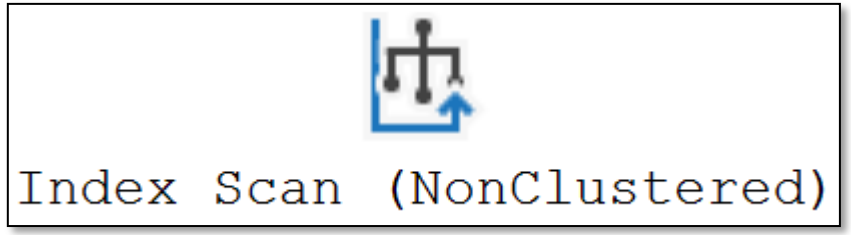
Total number of rows in the heap table
(at the time the query was last compiled)





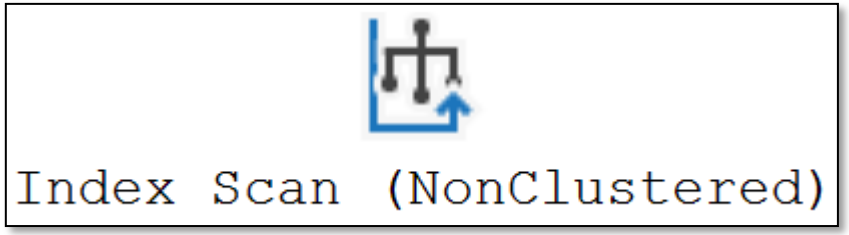
Scan operators

- Index Scan
  - Sometimes called Nonclustered Index Scan for clarity
  - Reads (all) data from a nonclustered index
    - On-disk rowstore
    - Columnstore (different name and icon used)
    - Memory-optimized (Bw-tree)
    - Memory-optimized Hash



Scan operators

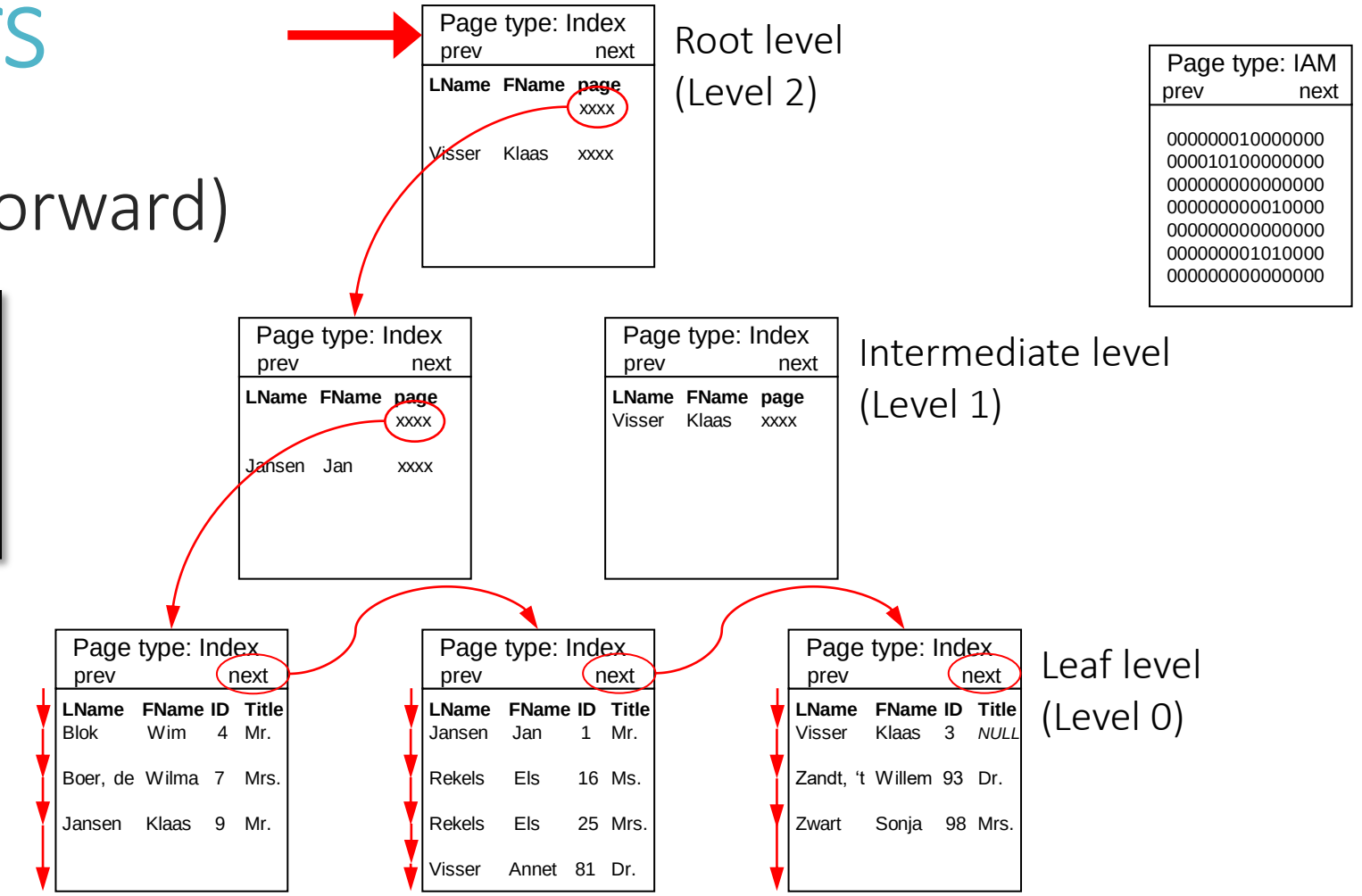
- Index Scan
  - Sometimes called Nonclustered Index Scan for clarity
  - Reads (all) data from a nonclustered index
  - Three processing modes
    - Ordered forward
    - Ordered backward
    - Unordered



Scan operators

Index Scan (ordered forward)

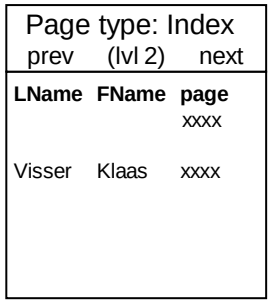
```
SELECT    Title,
          FirstName,
          LastName
FROM      dbo.PersonsCluster
ORDER BY  LastName;
```

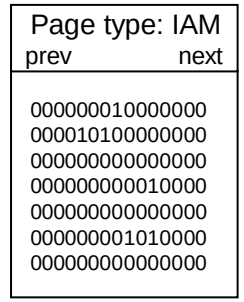


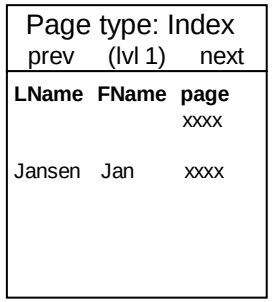
Scan operators

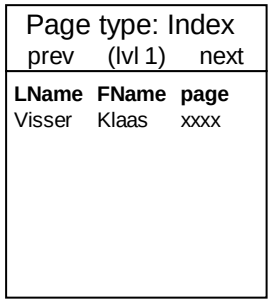
Index Scan (ordered forward)

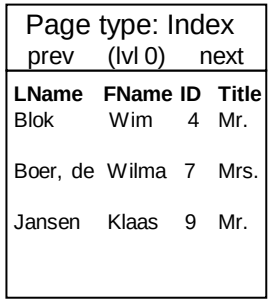
```
SELECT   Title,
         FirstName,
         LastName
FROM     dbo.PersonsCluster
ORDER BY LastName;
```

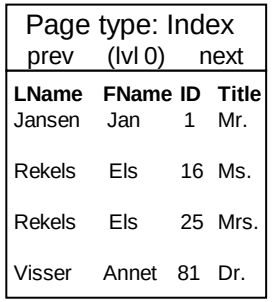


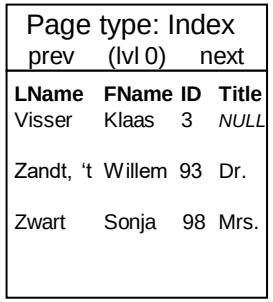












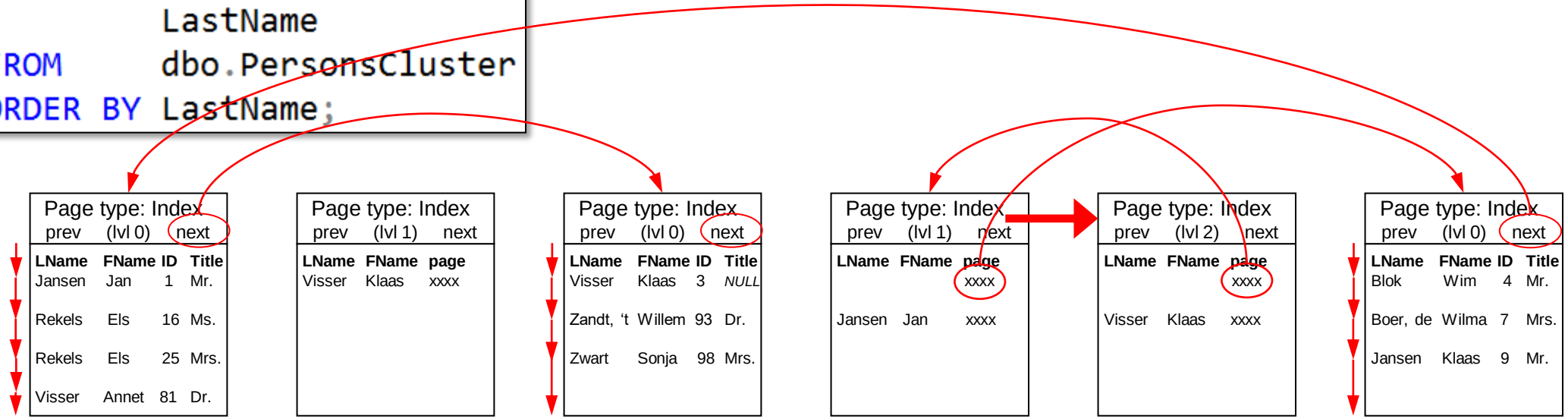
Scan operators

Index Scan (ordered forward)

Page type: IAM
prev next

000000010000000
000010100000000
000000000000000
000000000010000
000000000000000
000000001010000
000000000000000

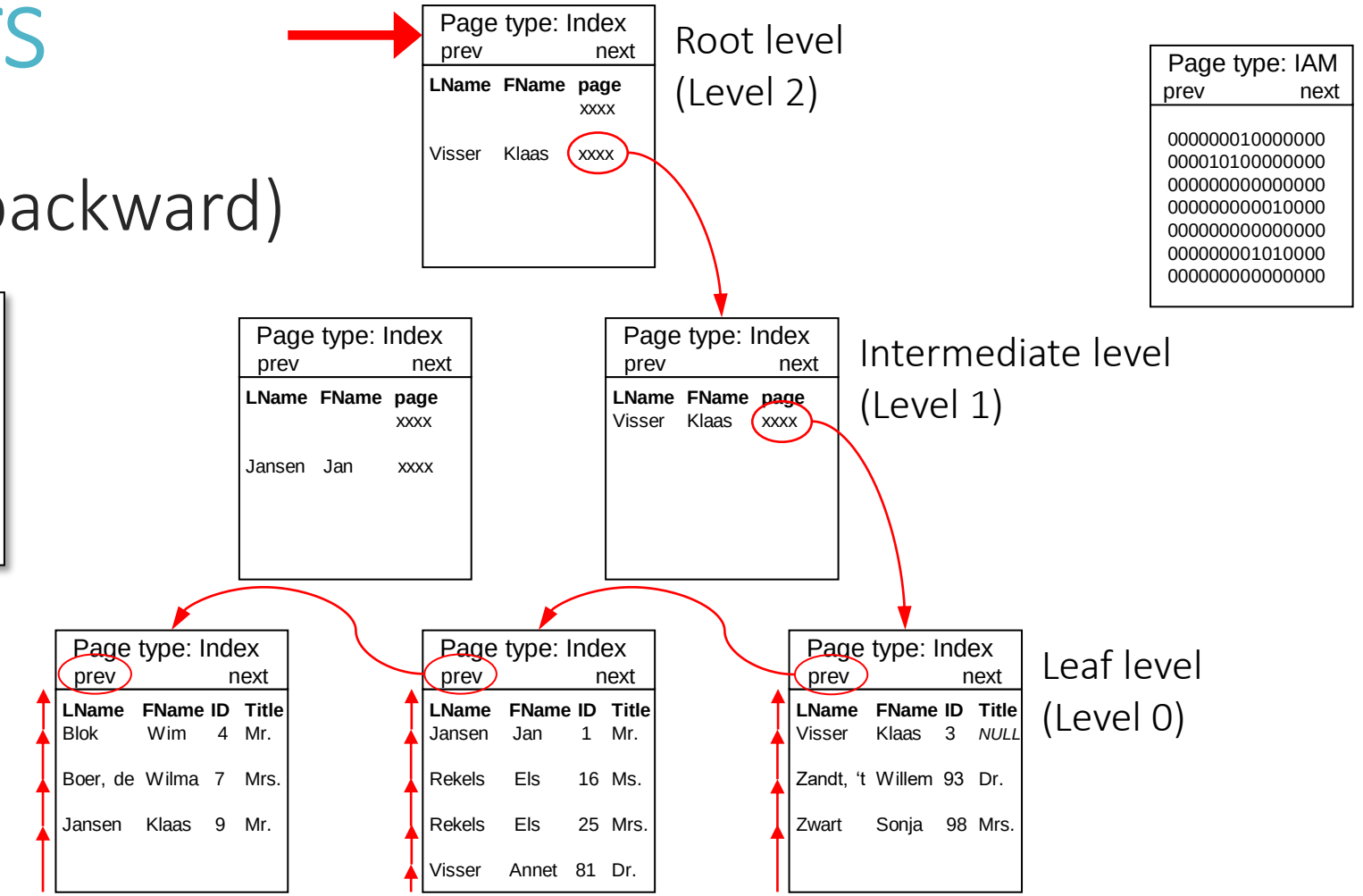
```
SELECT    Title,
          FirstName,
          LastName
FROM      dbo.PersonsCluster
ORDER BY LastName;
```



Scan operators

Index Scan (ordered backward)

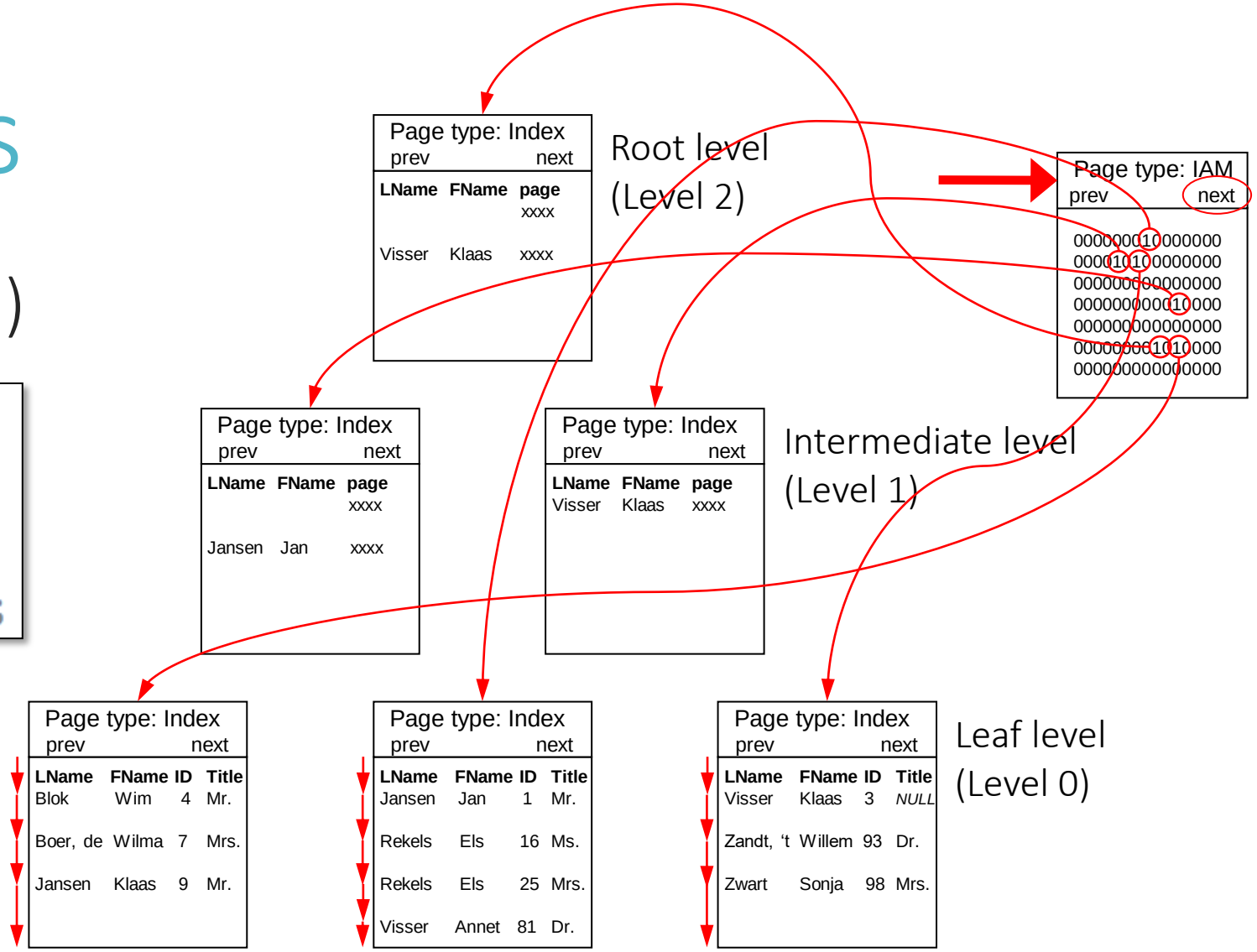
```
SELECT    LastName,
          FirstName
FROM      dbo.PersonsCluster
ORDER BY  LastName DESC,
          FirstName DESC;
```



Scan operators

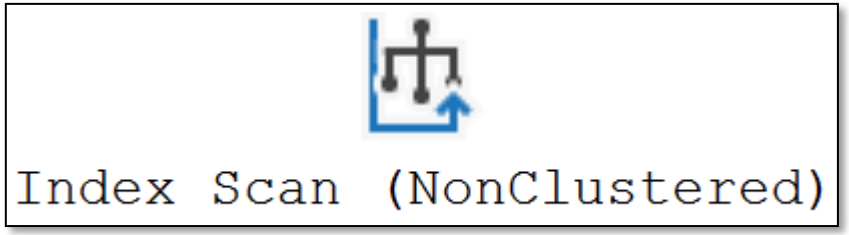
Index Scan (unordered)

```
SELECT PersonID,
       LastName,
       FirstName
FROM   dbo.PersonsCluster
                WITH (TABLOCK);
```



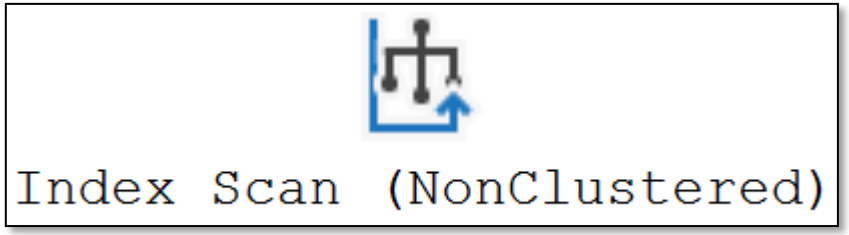
Scan operators

- Index Scan (unordered vs ordered)
  - Unordered scan touches all non-leaf pages
    - Just a tiny fraction of the total number of pages



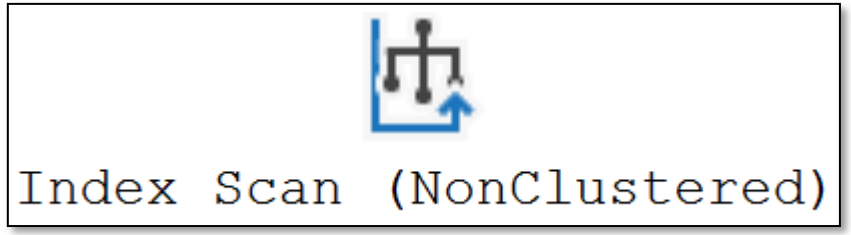
Scan operators

- Index Scan (unordered vs ordered)
  - Unordered scan touches all non-leaf pages
  - Unordered scan is sequential (file allocation order)
    - Larger read-ahead I/O requests
    - Less head movement (spinning disk)



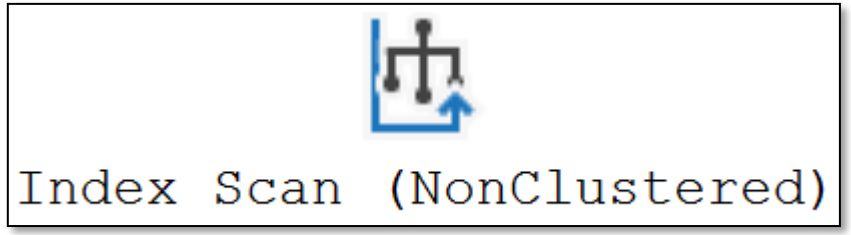
Scan operators

- Index Scan (unordered vs ordered)
  - Unordered scan touches all non-leaf pages
  - Unordered scan is sequential (file allocation order)
  - Determined by *Ordered* property
    - Set to True when ordered data is required
    - Set to False when order is irrelevant
      - Defers scan type choice to run time



Scan operators

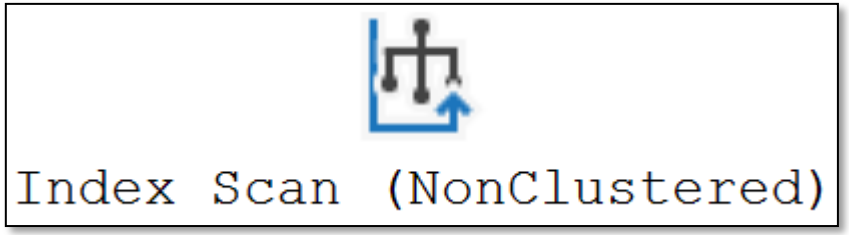
- Index Scan (unordered vs ordered)
  - Unordered scan touches all non-leaf pages
  - Unordered scan is sequential (file allocation order)
  - Determined by *Ordered* property
  - Unordered scan is subject to concurrency problems
    - Only permitted when …
      - Table-level lock used
      - Index is read only
      - NOLOCK / READUNCOMMITTED



Scan operators

Index Scan (unordered vs ordered)

- Unordered scan touches all non-leaf pages
- Unordered scan is sequential (file allocation order)
- Determined by *Ordered* property
- Unordered scan is subject to concurrency problems
- Unordered scan considered too expensive for less than 64 pages



Scan operators

Index Scan

Properties (popup)

Ordered scan required?
(by optimizer)

Which index is scanned?

Index Scan (NonClustered)

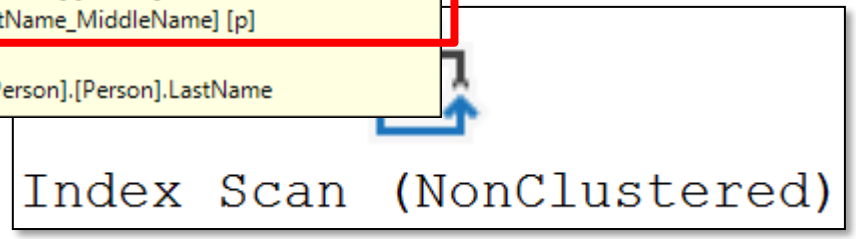
Scan a nonclustered index, entirely or only a range.

Physical Operation	Index Scan
Logical Operation	Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	19972
Actual Number of Rows for All Executions	103
Actual Number of Batches	0
Estimated I/O Cost	0,080162
Estimated Operator Cost	0,102288 (91%)
Estimated CPU Cost	0,0221262
Estimated Subtree Cost	0,102288
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows Per Execution	109,003
Estimated Number of Rows to be Read	19972
Estimated Row Size	35 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	2

Predicate
[AdventureWorks2017].[Person].[Person].[FirstName] as [p].[FirstName]=N'Richard'

Object
[AdventureWorks2017].[Person].[Person].[IX_Person_LastName_FirstName_MiddleName] [p]

Output List
[AdventureWorks2017].[Person].[Person].LastName



Scan operators

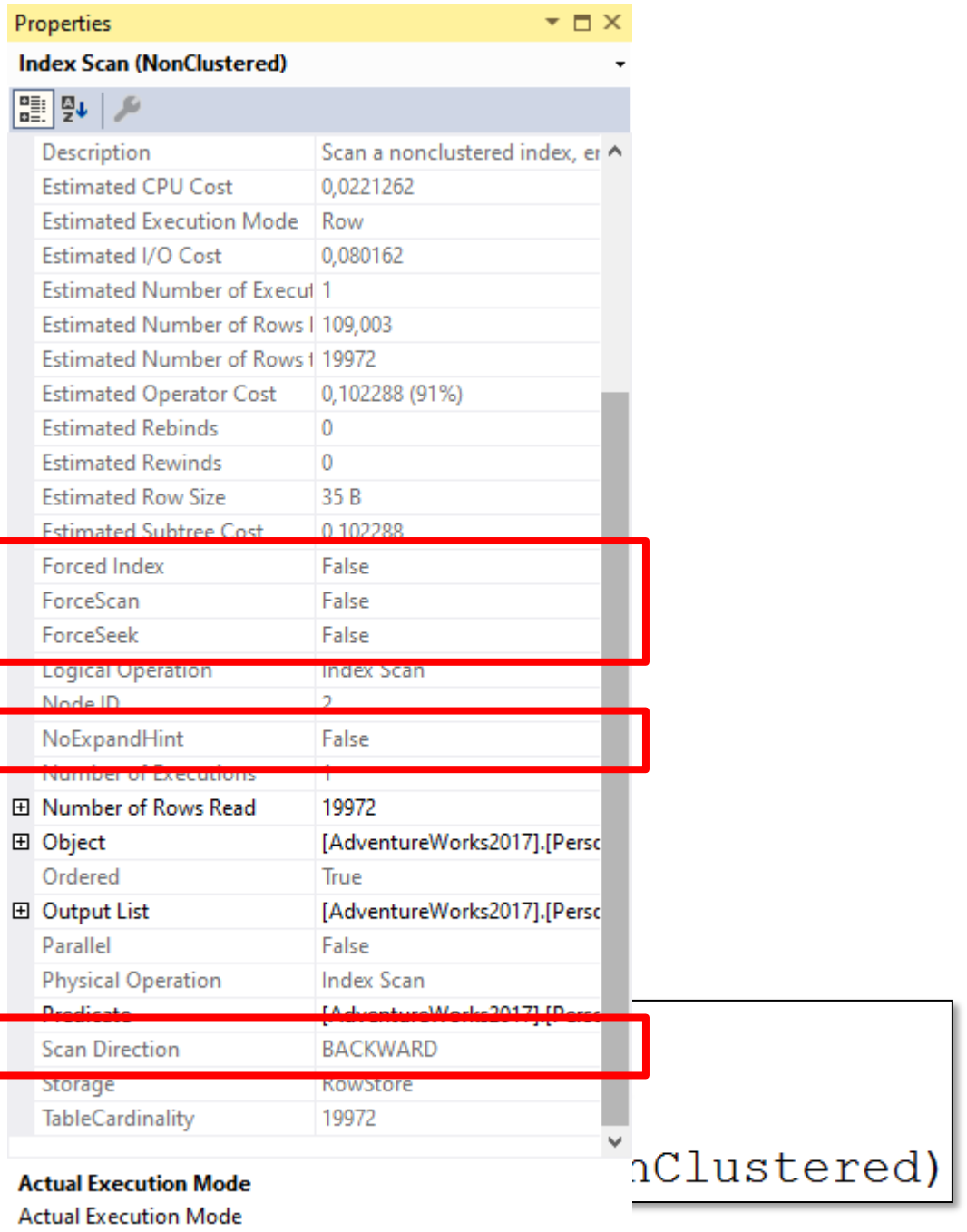
Index Scan

Properties (full list)

Were various hints used in the query?

Indexed view with NOEXPAND hint

Order scan: forward or backward?



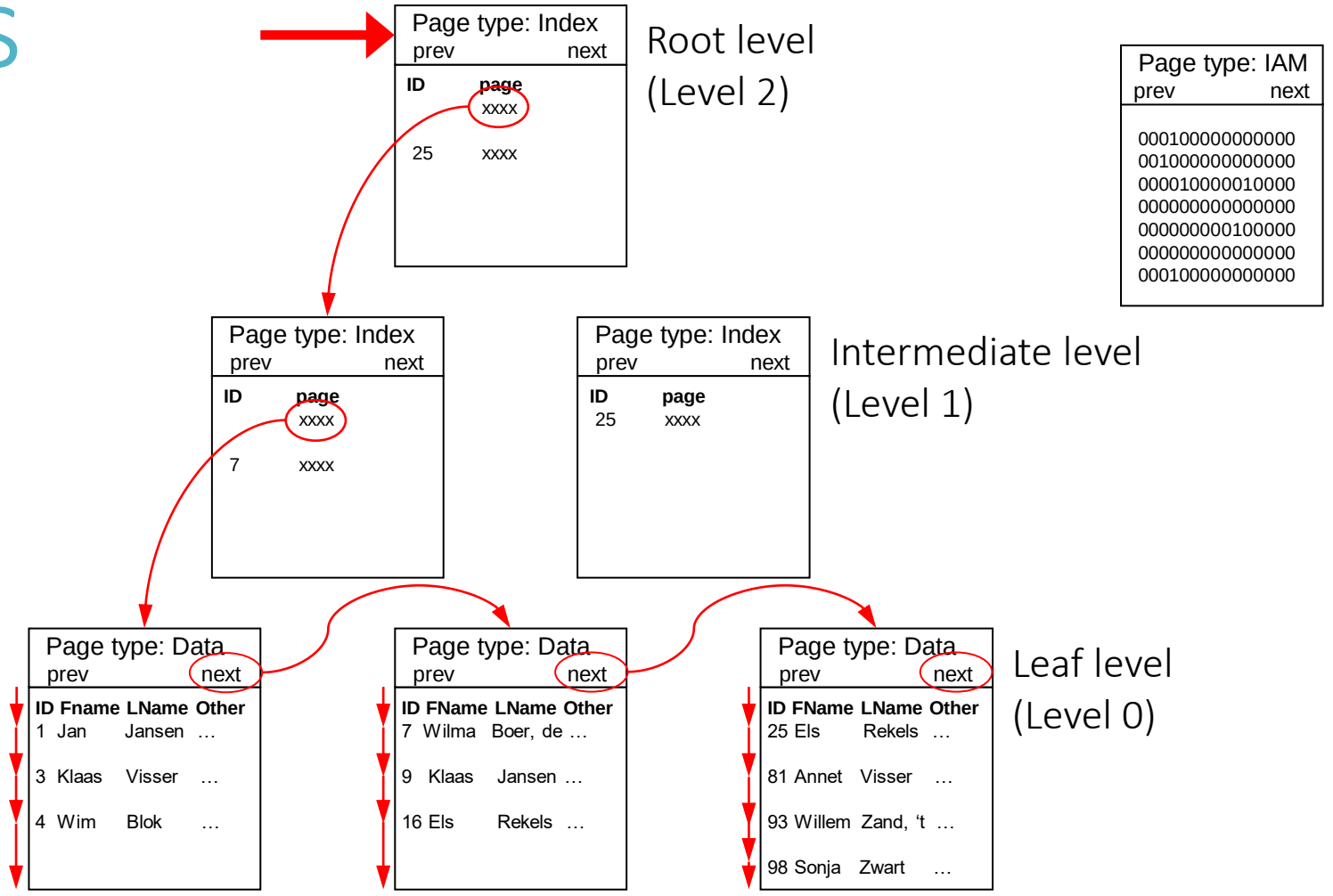
Scan operators

- Clustered Index Scan
  - Actually same as (nonclustered) Index Scan
  - Reads (all) data from a ***clustered*** index
    - On-disk rowstore
    - Columnstore (different name and icon used)
    - Memory-optimized (Bw-tree)
    - Memory-optimized Hash

Scan operators

Clustered Index Scan

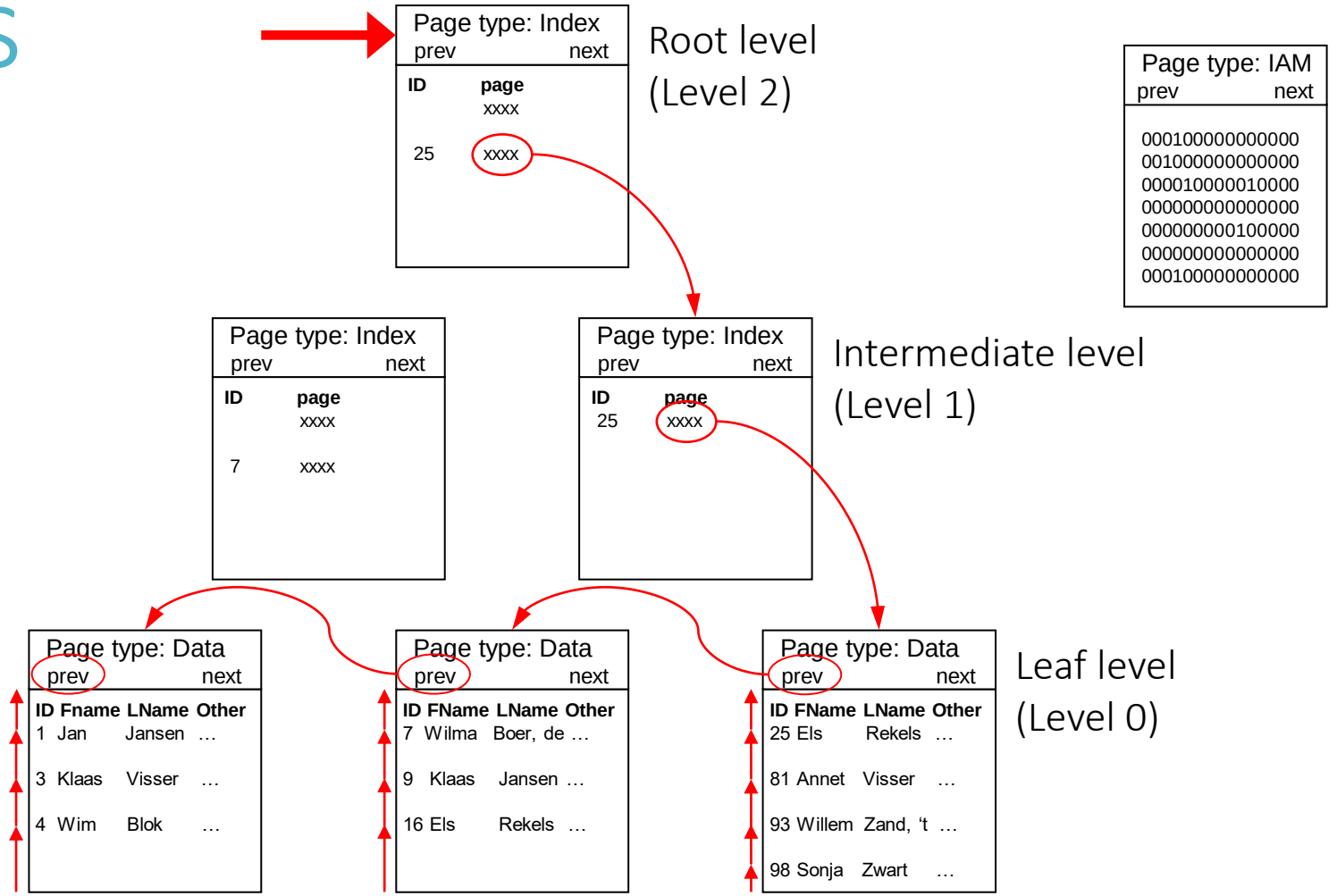
Ordered forward



Scan operators

Clustered Index Scan

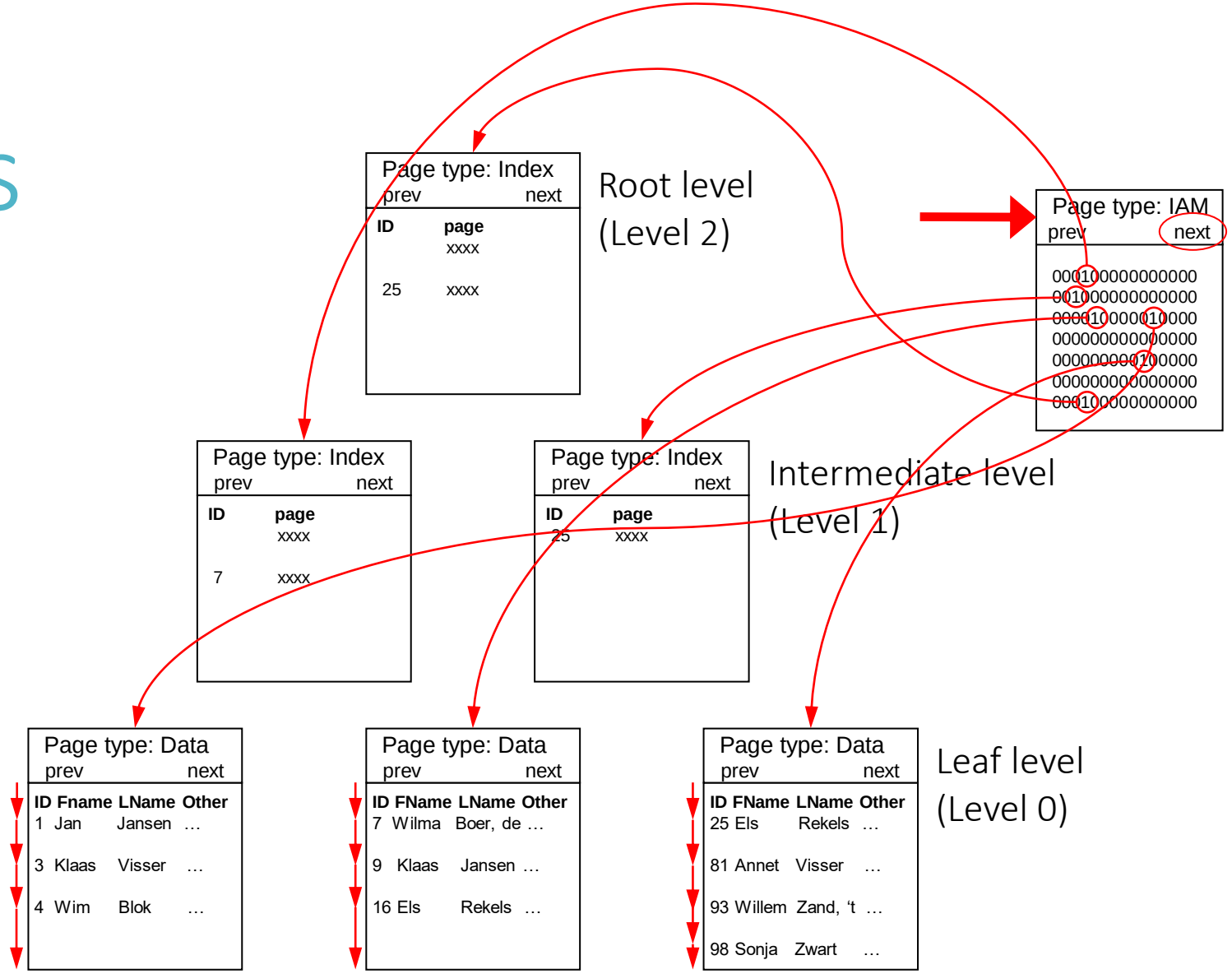
- Ordered forward
- Ordered backward



Scan operators

Clustered Index Scan

- Ordered forward
- Ordered backward
- Unordered (IAM scan)



Scan operators

- Clustered Index Scan
  - Actually same as (nonclustered) Index Scan
  - Reads (all) data from a ***clustered*** index
  - Can access ALL data
    - No need for Lookup to get additional data
  - Needs to read more pages

Summary

- Scan operators
  - Table Scan
    - IAM scan only
  - Index Scan / Clustered Index Scan
    - Ordered scan or IAM scan
  - Predicate
    - Pushed down to eliminate rows early

Next chapters

Chapter 3: Seek operators

- Index Seek
- Clustered Index Seek

Chapter 4: Lookup operators

Chapter 5: Special scans