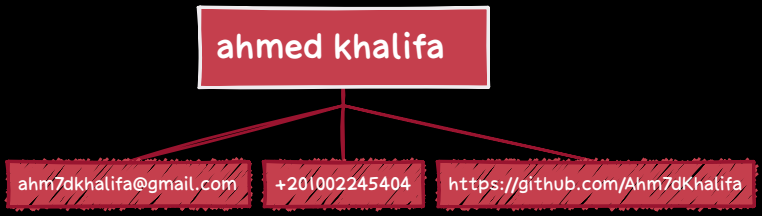
Physical Database Design

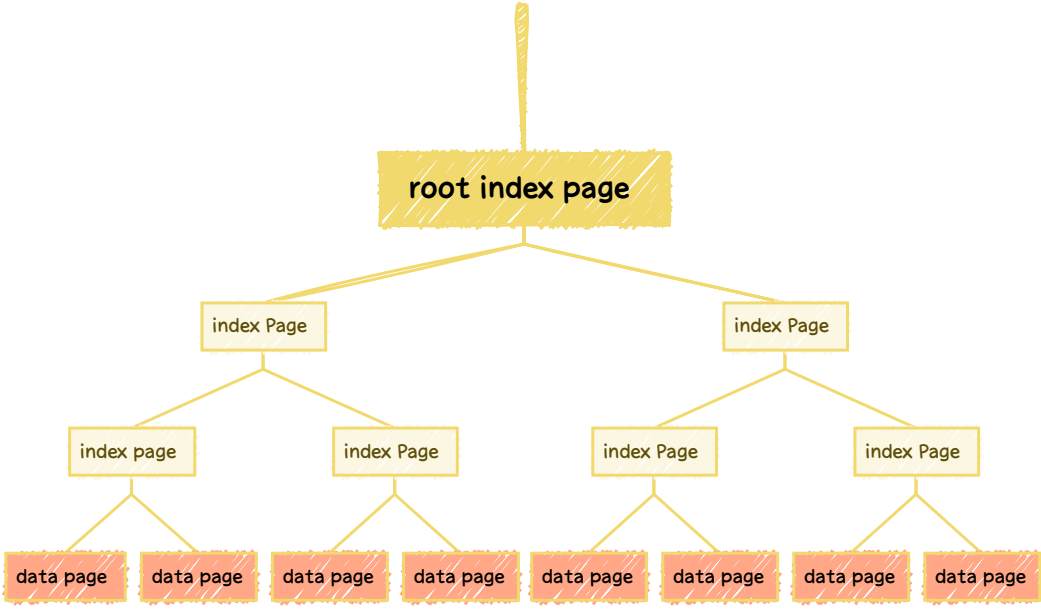
Indexes Part 3 : Clustered and non-clustered Indexes

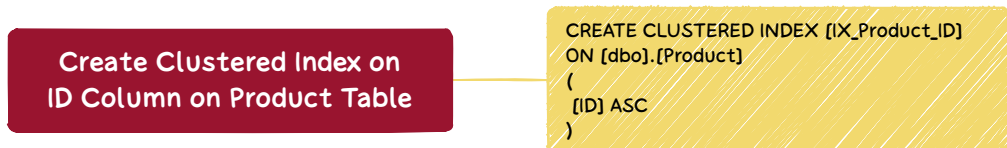


Clustered Index

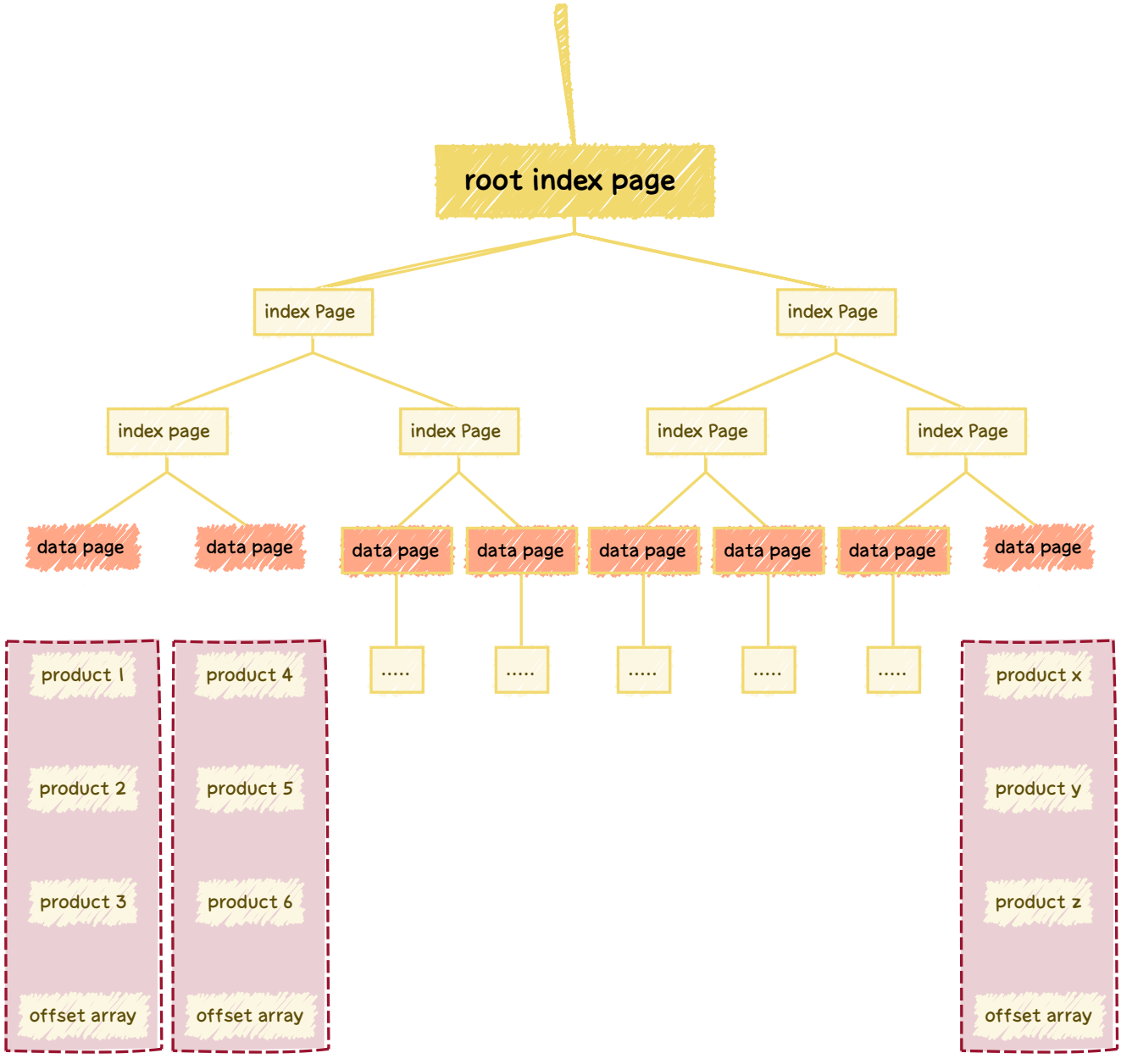
- in-disk b-tree data structure consist of 3 levels :
  - 1- root node : contain
    - clustered index keys
    - with address of pages of other lower index pages
  - 2- intermediate nodes : contain
    - clustered index keys
    - with address of pages of other lower index pages
    - or address of actual data pages
  - 3- leaf nodes : contain
    - actual data pages
- data is stored on disk with order
  - Index Key Is Sorted in Indexes Pages in Root or intermediate nodes
  - rows is Sorted (Logically or Physically) in Actual Data Pages in last level

in-disk b-tree data structure





Product Cluster Index Table



Clustered Index On ID Column

First Level : Root Node

page 1000

index key	page id
30	1001
50	1002
.....	.....
z	100X

Second Level : intermediate nodes

page 1001

index key	page id
12	1003
20	1004
.....	.....

page 1002

index key	page id
35	1005
45	1006
.....	.....

index key	page id
....	.....
....	.....
Z	.....

page 1003

index key	page id
3	7001
6	7002
.....	.....

page 1004

index key	page id
.....	.....
.....	.....
.....	.....

index key	page id
....	.....
....	.....
.....	.....

index key	page id
.....	....
.....	....
.....	.....

index key	page id
.....	.....
.....	.....
.....	.....

index key	page id
.....	.....
.....	.....
.....	.....

Third Level : Leaf Nodes

page 7001

ID	name	price	Qty
1	C2	...	...
2	D1	...	...
3	A2	...	...

page 7002

ID	name	price	Qty
4	B1	...	...
5	A3	...	...
6	C1	...	...

page 7003

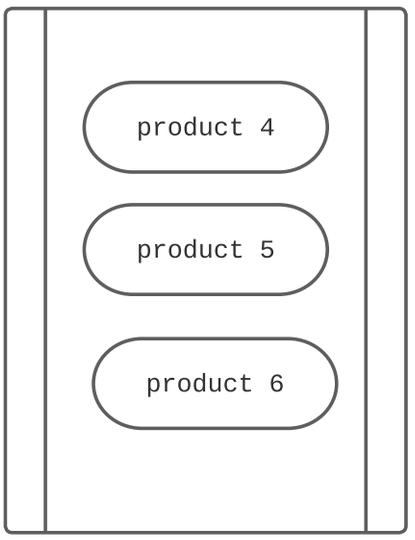
ID	name	price	Qty
7	D2	...	...
8	B2	...	...
9	A1	...	...

Logical Order vs Physical Order

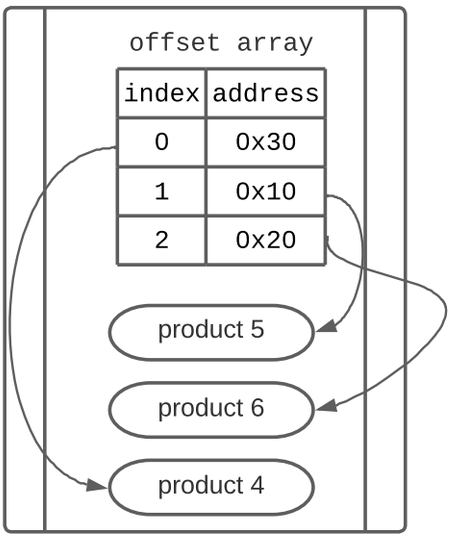
when insert unsorted data to clustered index table and all rows will be stored on the same page Like :

```
Insert into [dbo].[Product] Values ( 5 , 'Product 5' , 100 , 20 );
Insert into [dbo].[Product] Values ( 6, 'Product 6' , 120, 17);
Insert into [dbo].[Product] Values ( 4, 'Product 4' , 50, 26);
```

you think the data will be stored with physical order in the same page like :

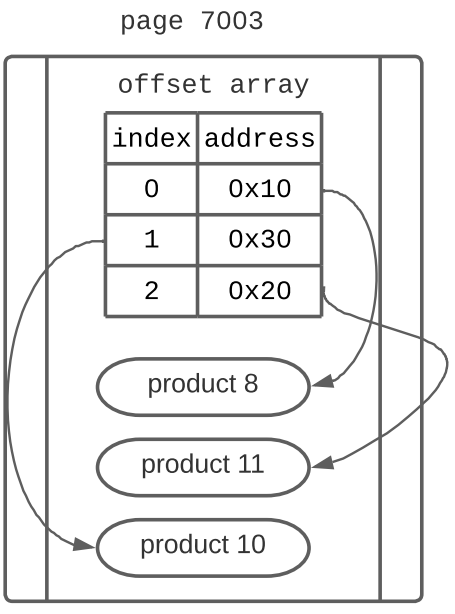


but actually database engine will stored them as they insert first , then will sort or logically order them by offset array and the data will be appear as they ordered physically.



page splits

assume we have this page

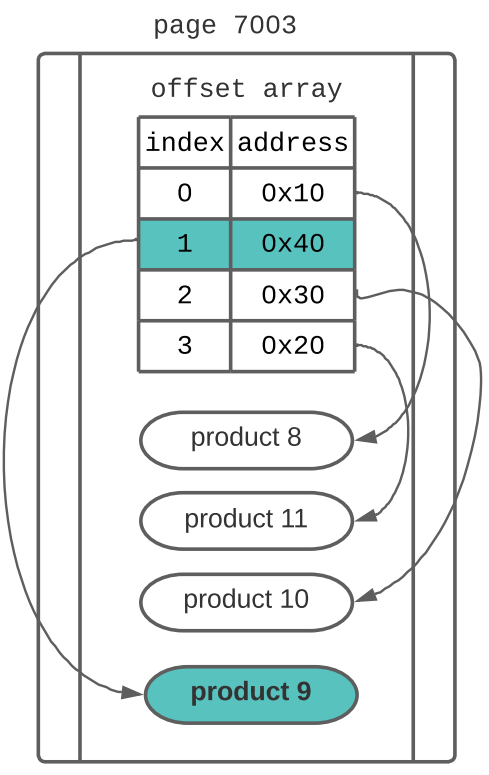



```
Insert into [dbo].[Product] Values ( 9 , 'Product 9' , 170 , 90 );
```

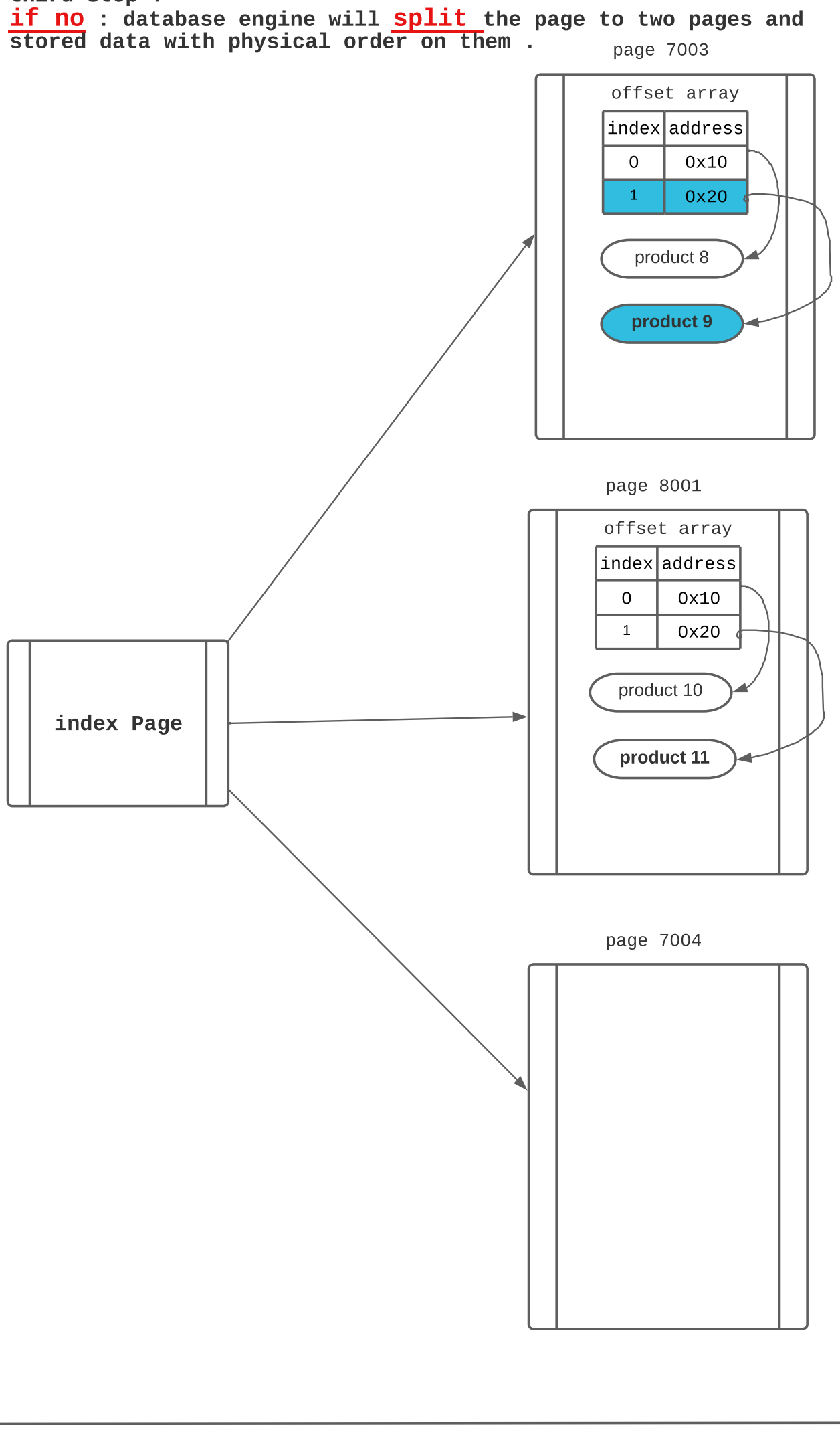
how database engine will insert this row physically ?

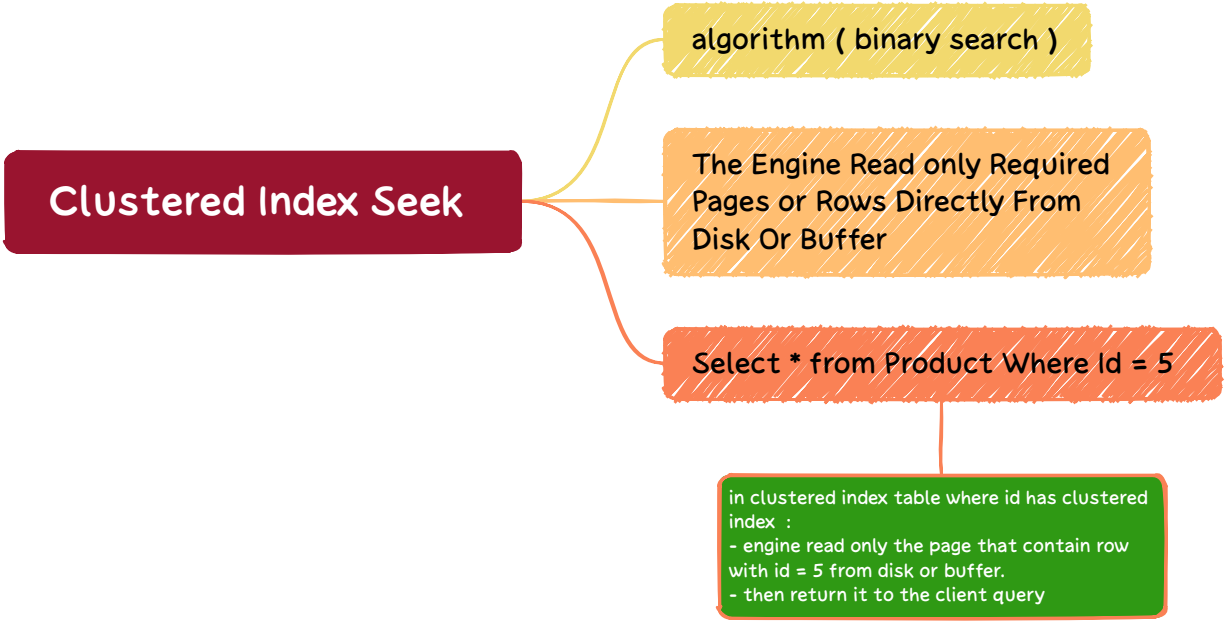
first step :
database engine will determine the data page which the row must be inserted to it to keep the order , then database engine will check if there available space to insert this row at this page ?

second step :
if yes : database engine will attach this row at the end of data page and refactor offset array to maintenance order of data



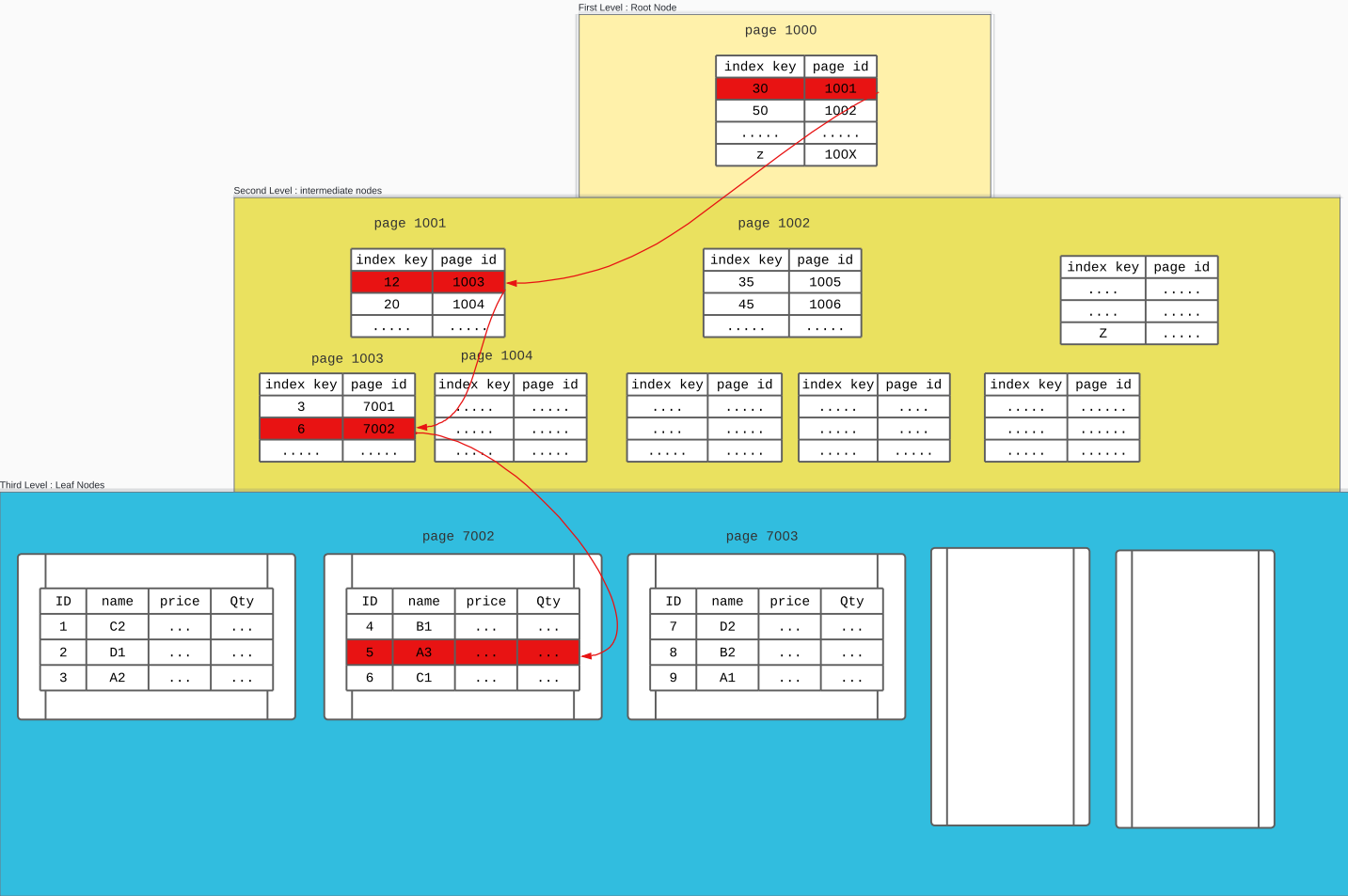
third step :
if no : database engine will split the page to two pages and stored data with physical order on them .





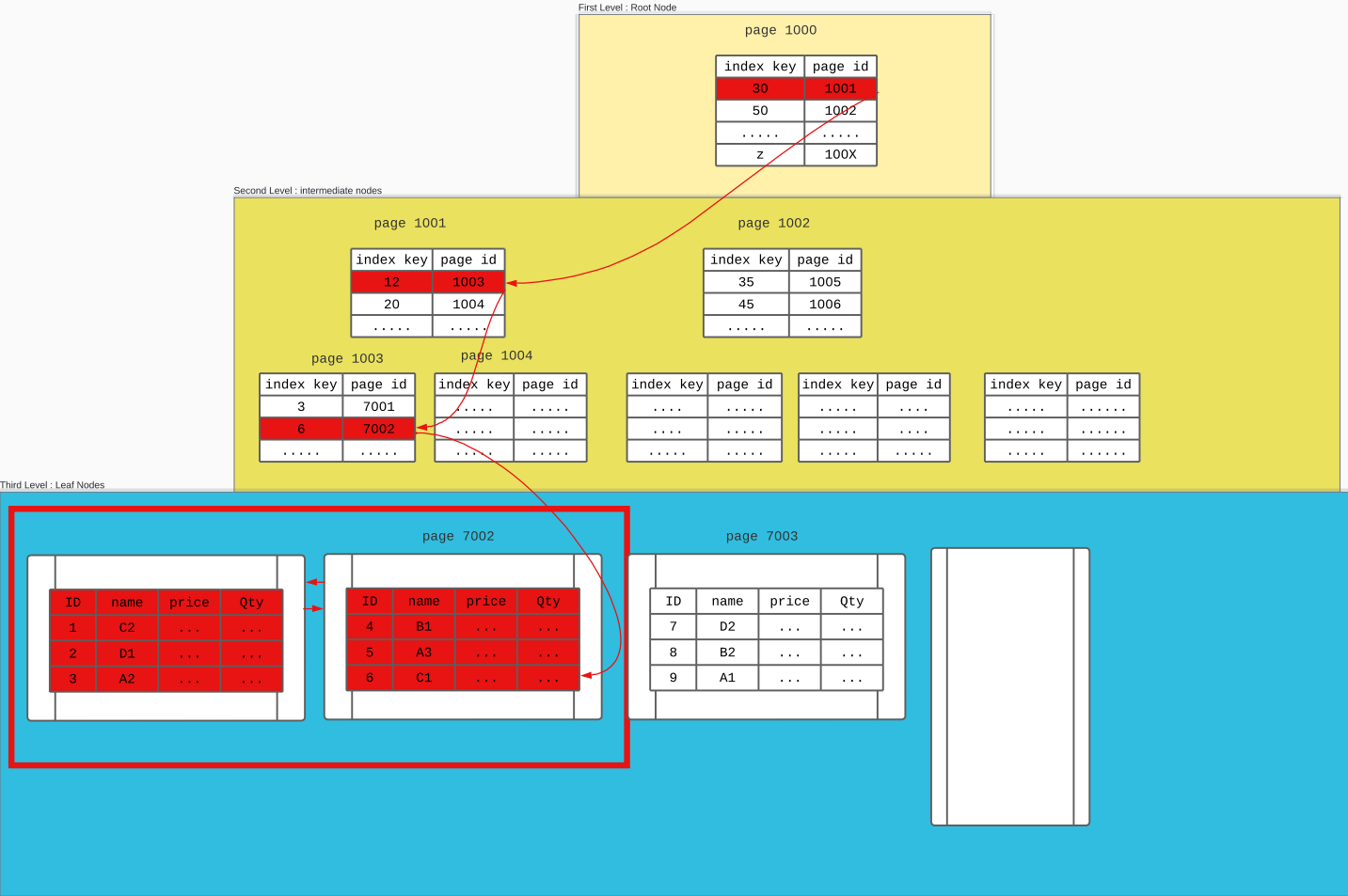
Clustered Index Seek Example :
SELECT * FROM Product WHERE ID = 5

Clustered Index On ID Column



Clustered Index Seek Example :
SELECT * FROM Product WHERE ID <= 6

Clustered Index On ID Column



Clustered Index Seek Complexity

Algorithm : Binary Search

Complexity ~=
number of required indexes pages (b-tree height)
+
number of required data pages (actual data pages in last level)

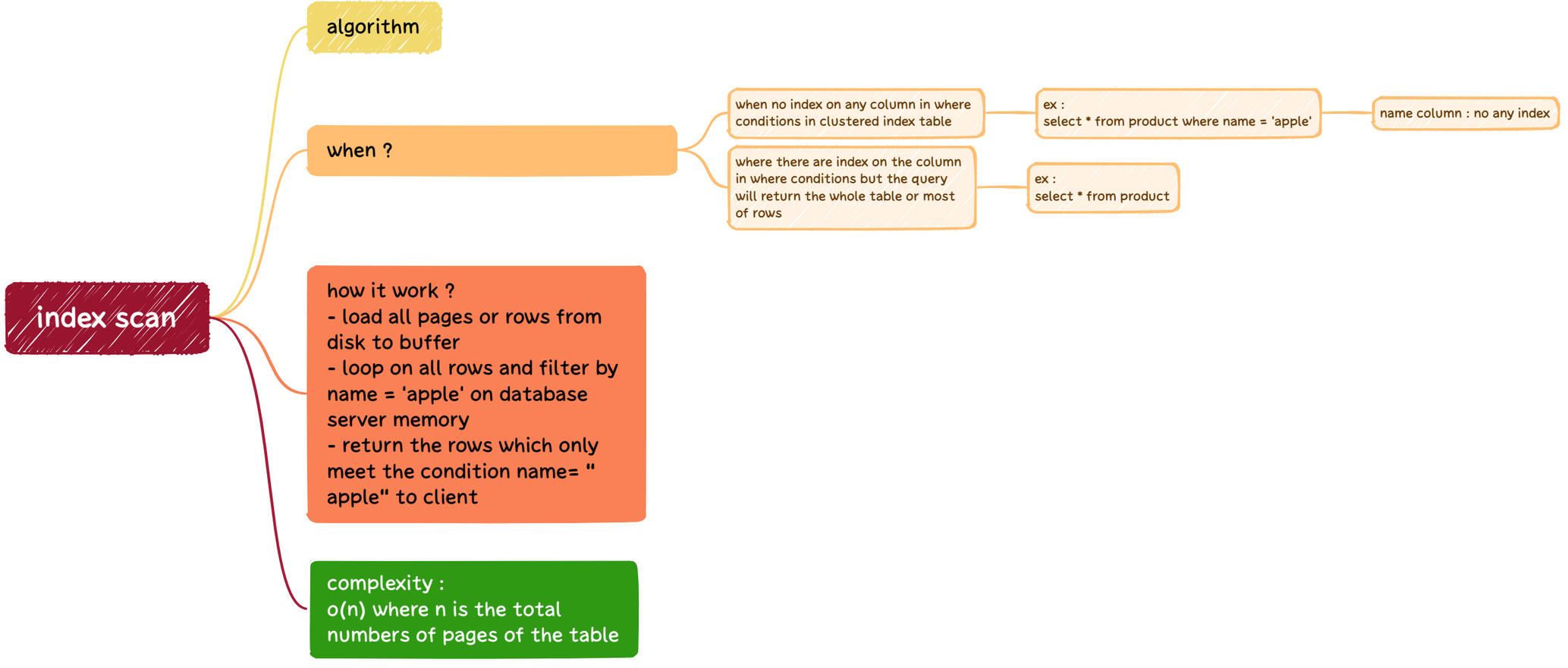
why page is the unit ?
page is the basic unit for the database engine not row.
if the page contain multiple row and we want only to return one row , the engine will read the whole page from disk and filter on database server memory.

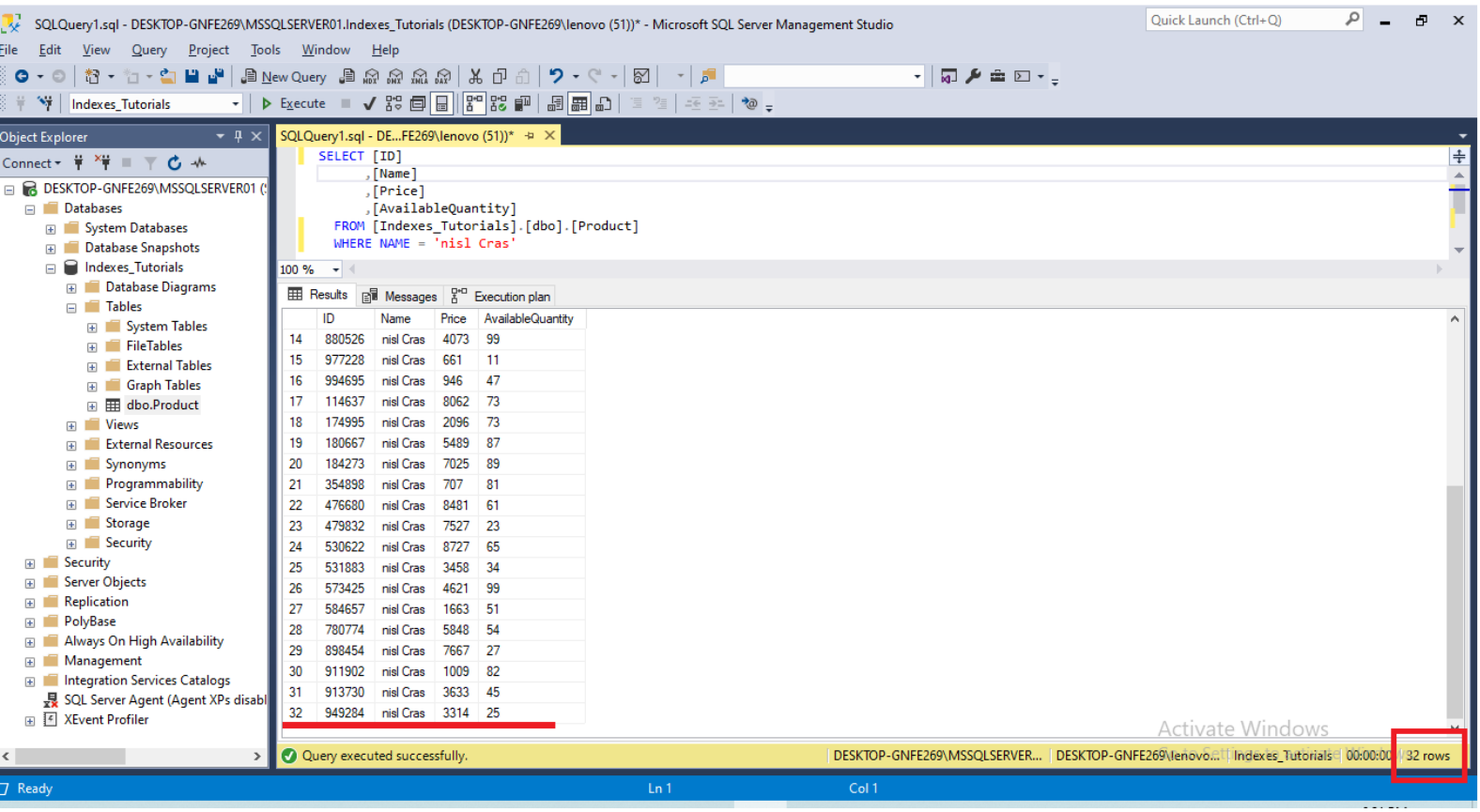
for ex :
Select * from product where id = 5

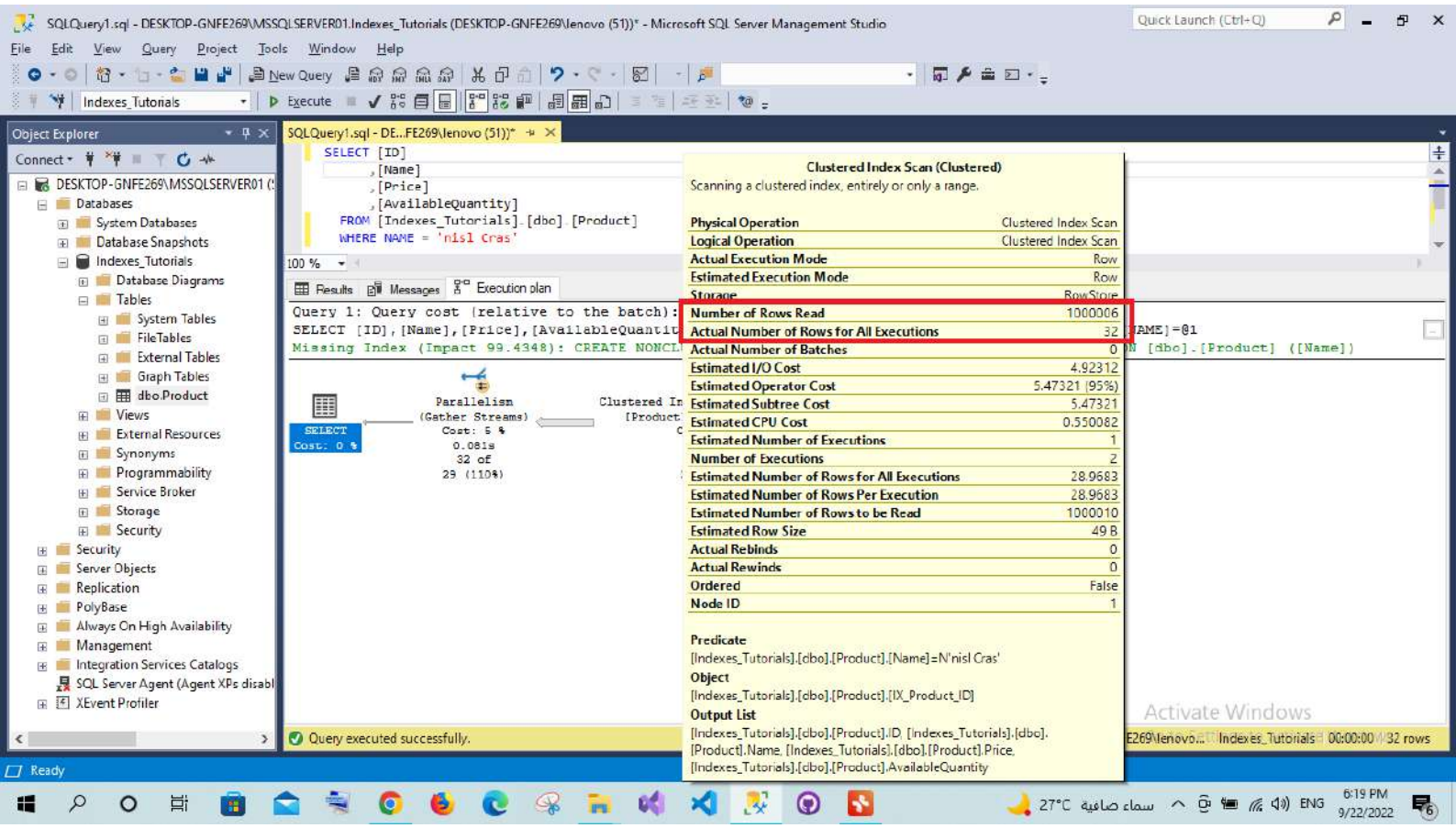
- number of indexes pages : 3
- number of actual data page : 1
- total : 4 pages

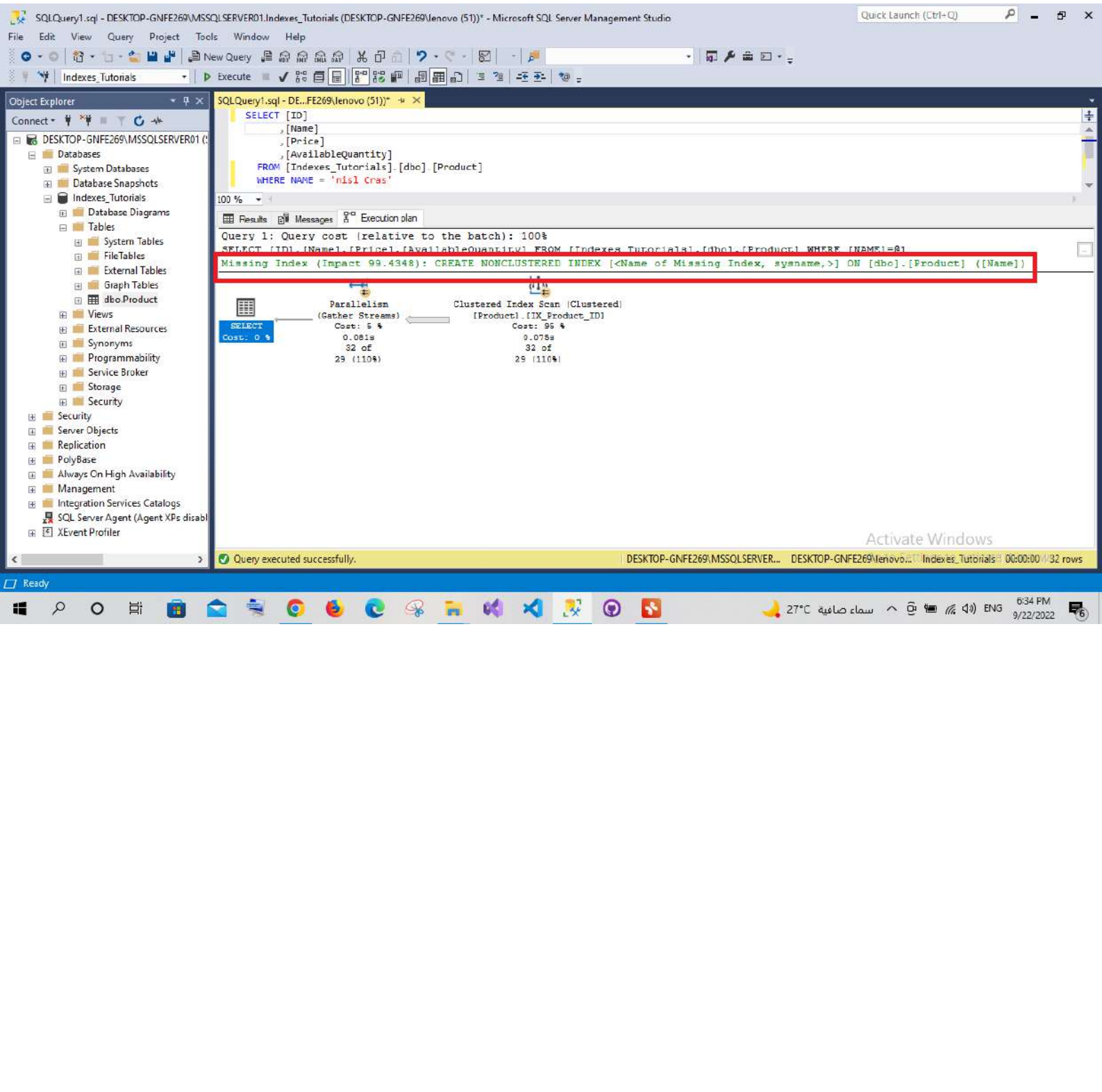
for ex :
select * from product where id < 7

- number of indexes pages : 3
- number of actual data page : 2
- total : 5 pages




SQLQuery1.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (51))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Indexes_Tutorials
Execute
Object Explorer
Connect
DESKTOP-GNFE269\MSSQLSERVER01
Databases
System Databases
Database Snapshots
Indexes_Tutorials
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Product
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - DE...FE269\lenovo (51))*
SELECT [ID]
,[Name]
,[Price]
,[AvailableQuantity]
FROM [Indexes_Tutorials].[dbo].[Product]
WHERE NAME = 'nisl Cras'
100 %
Results
Messages
Execution plan
ID Name Price AvailableQuantity
14 880526 nisl Cras 4073 99
15 977228 nisl Cras 661 11
16 994695 nisl Cras 946 47
17 114637 nisl Cras 8062 73
18 174995 nisl Cras 2096 73
19 180667 nisl Cras 5489 87
20 184273 nisl Cras 7025 89
21 354898 nisl Cras 707 81
22 476680 nisl Cras 8481 61
23 479832 nisl Cras 7527 23
24 530622 nisl Cras 8727 65
25 531883 nisl Cras 3458 34
26 573425 nisl Cras 4621 99
27 584657 nisl Cras 1663 51
28 780774 nisl Cras 5848 54
29 898454 nisl Cras 7667 27
30 911902 nisl Cras 1009 82
31 913730 nisl Cras 3633 45
32 949284 nisl Cras 3314 25
Activate Windows
Query executed successfully.
DESKTOP-GNFE269\MSSQLSERVER...
DESKTOP-GNFE269\lenovo...
Indexes_Tutorials
00:00:00
32 rows
Ready
Ln 1
Col 1


SQLQuery1.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (51))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Indexes_Tutorials
Execute
Object Explorer
Connect
DESKTOP-GNFE269\MSSQLSERVER01
Databases
System Databases
Database Snapshots
Indexes_Tutorials
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Product
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - DE..FE269\lenovo (51))*
SELECT [ID]
,[Name]
,[Price]
,[AvailableQuantity]
FROM [Indexes_Tutorials].[dbo].[Product]
WHERE NAME = 'nisl Cras'
100 %
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch):
SELECT [ID],[Name],[Price],[AvailableQuantit
Missing Index (Impact 99.4348): CREATE NONCL
SELECT
Cost: 0 %
Parallelism
(Gather Streams)
Cost: 5 %
0.081s
32 of
29 (110%)
Clustered In
[Product
Clustered Index Scan (Clustered)
Scanning a clustered index, entirely or only a range.
Physical Operation Clustered Index Scan
Logical Operation Clustered Index Scan
Actual Execution Mode Row
Estimated Execution Mode Row
Storage RowStore
Number of Rows Read 1000006
Actual Number of Rows for All Executions 32
Actual Number of Batches 0
Estimated I/O Cost 4.92312
Estimated Operator Cost 5.47321 (95%)
Estimated Subtree Cost 5.47321
Estimated CPU Cost 0.550082
Estimated Number of Executions 1
Number of Executions 2
Estimated Number of Rows for All Executions 28.9683
Estimated Number of Rows Per Execution 28.9683
Estimated Number of Rows to be Read 1000010
Estimated Row Size 49 B
Actual Rebinds 0
Actual Rewinds 0
Ordered False
Node ID 1
Predicate
[Indexes_Tutorials].[dbo].[Product].[Name]=N'nisl Cras'
Object
[Indexes_Tutorials].[dbo].[Product].[IX_Product_ID]
Output List
[Indexes_Tutorials].[dbo].[Product].ID, [Indexes_Tutorials].[dbo].
[Product].Name, [Indexes_Tutorials].[dbo].[Product].Price,
[Indexes_Tutorials].[dbo].[Product].AvailableQuantity
Activate Windows
Query executed successfully.
Indexes_Tutorials 00:00:00 32 rows
Ready
27°C
ENG
6:19 PM
9/22/2022


SQLQuery1.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (51))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Indexes_Tutorials
Execute
Object Explorer
Connect
DESKTOP-GNFE269\MSSQLSERVER01
Databases
System Databases
Database Snapshots
Indexes_Tutorials
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Product
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - DE..FE269\lenovo (51))*
SELECT [ID]
,[Name]
,[Price]
,[AvailableQuantity]
FROM [Indexes_Tutorials].[dbo].[Product]
WHERE NAME = 'nisl Cras'
100 %
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [ID],[Name],[Price],[AvailableQuantity] FROM [Indexes_Tutorials].[dbo].[Product] WHERE [NAME]=@1
Missing Index (Impact 99.4348): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[Product] ([Name])
SELECT
Cost: 0 %
Parallelism
(Gather Streams)
Cost: 5 %
0.081s
32 of
29 (110%)
Clustered Index Scan (Clustered)
[Product].[IX_Product_ID]
Cost: 95 %
0.075s
32 of
29 (110%)
Activate Windows
Query executed successfully.
DESKTOP-GNFE269\MSSQLSERVER...
DESKTOP-GNFE269\lenovo...
Indexes_Tutorials
00:00:00
32 rows
Ready
27°C
ENG
6:34 PM
9/22/2022

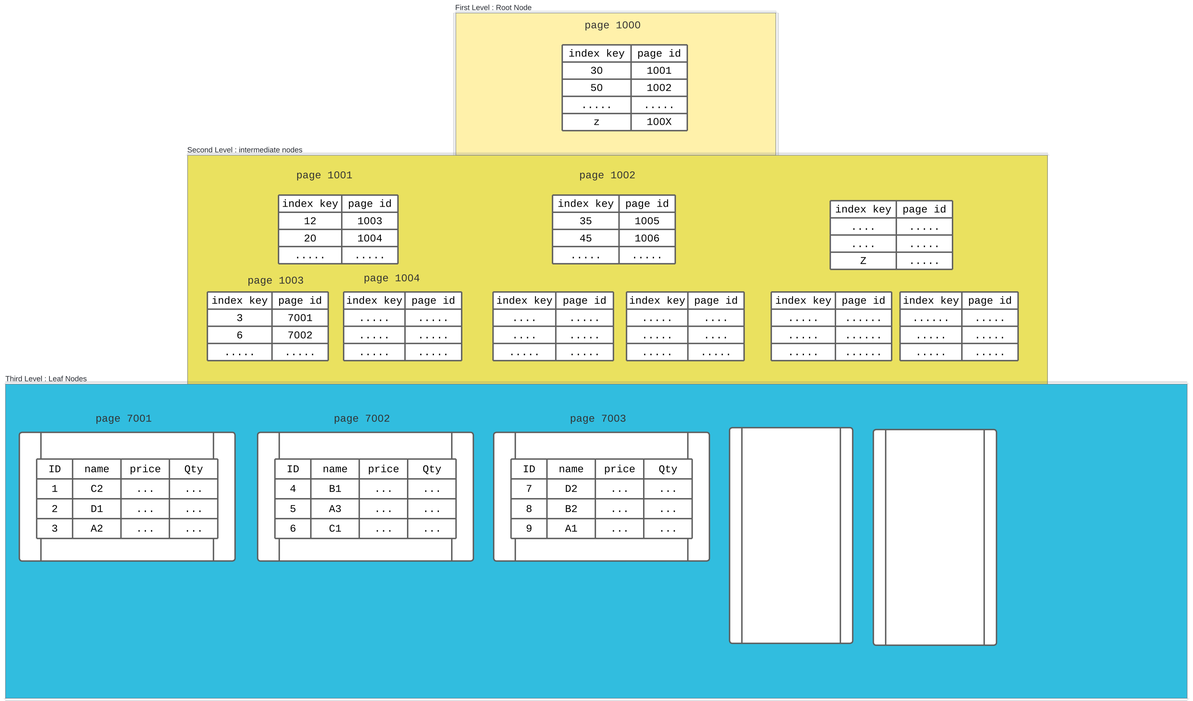
non-Clustered Index

- in-disk b-tree data structure consist of 3 levels :
  - 1- root node : contain
    - non-clustered index keys
    - with address of pages of other lower index pages
  - 2- intermediate nodes : contain
    - non-clustered index keys
    - with address of pages of other lower index pages
  - 3- leaf nodes : contain
    - pointers to actual data
- Nonclustered indexes have a structure separate from the data rows or actual table
- Indexes Key Is Sorted in Indexes Pages in Root or intermediate nodes
- can used with heap table or clustered table
- pointers :
  - in case clustered table : clustered index keys
  - in case heap table :
    ROWID (File + Page + row)
- table can contain many non-clustered indexes

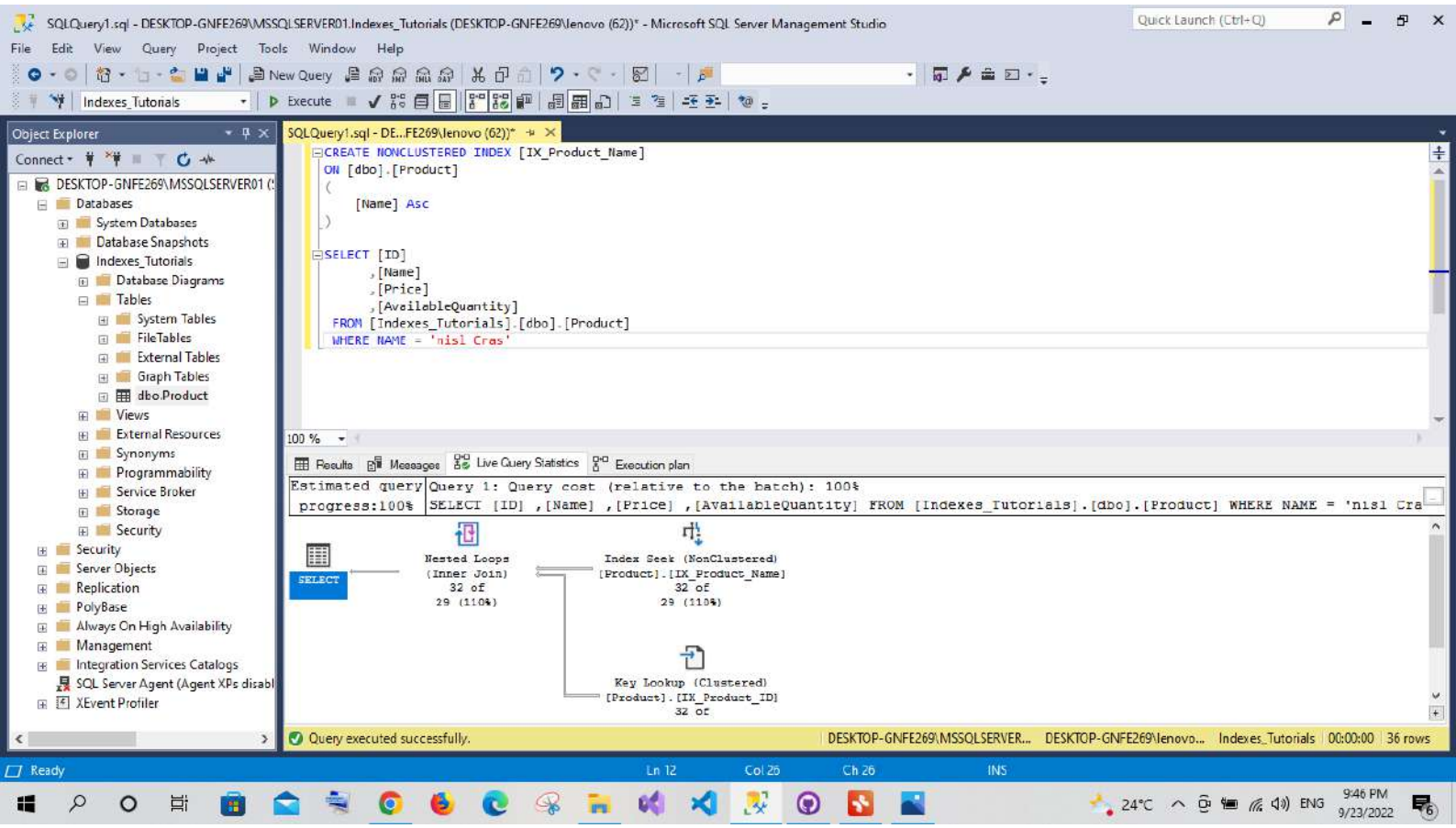
Create Non-Clustered Index on Name Column on Product Table

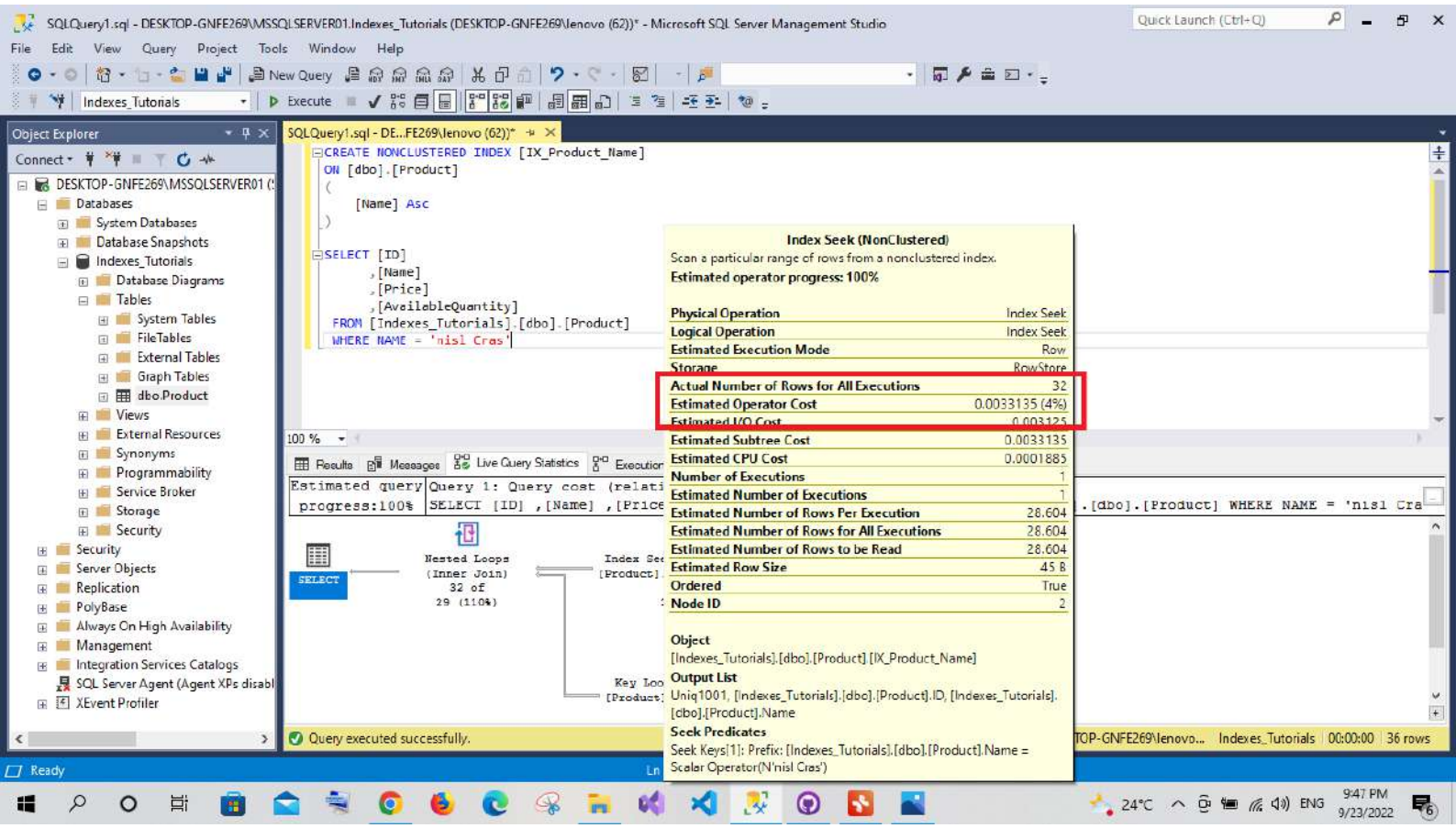
```
CREATE NonCLUSTERED
INDEX [IX_Product_Name]
ON [dbo].[Product]
(
 [Name] Asc
)
```

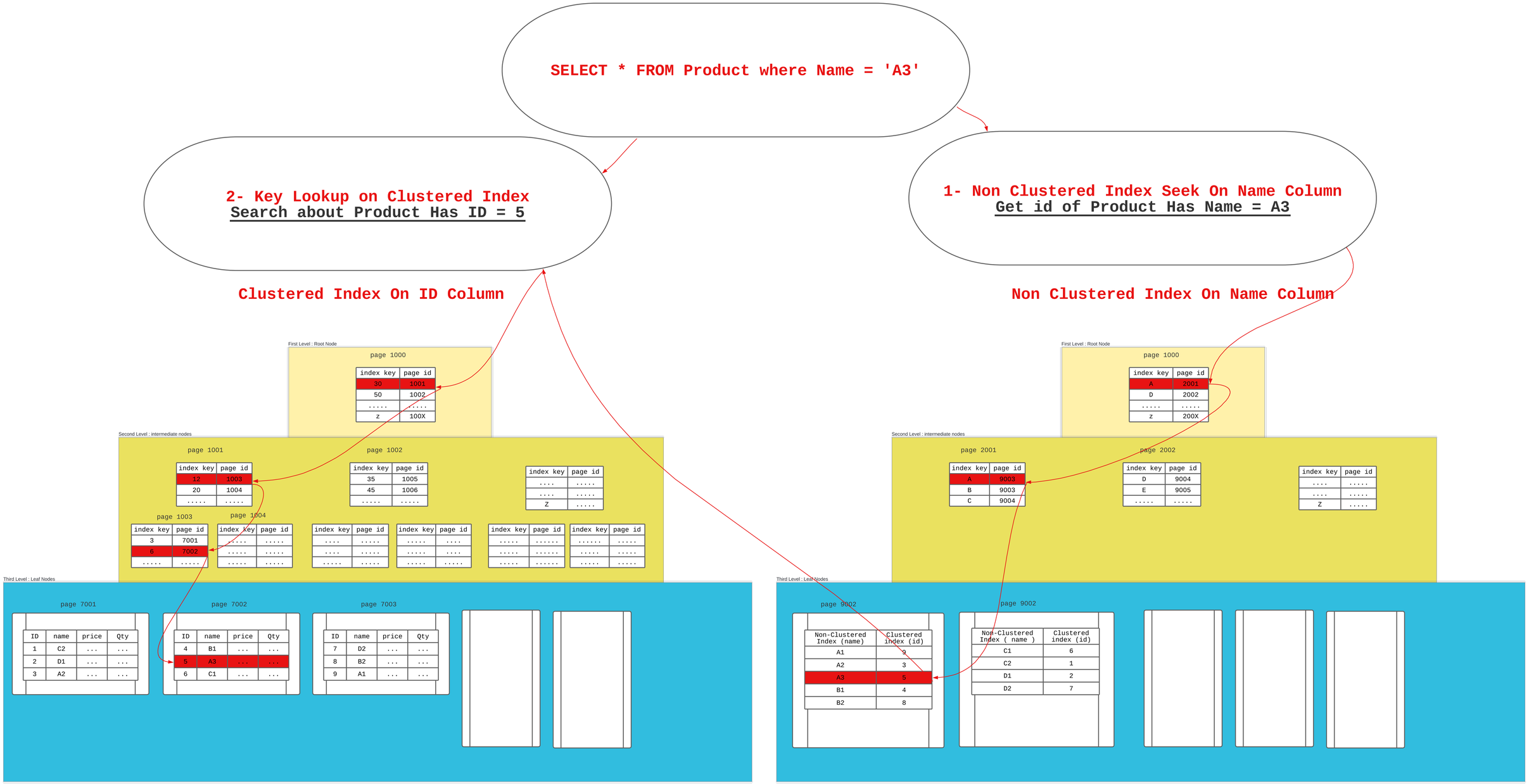
Clustered Index On ID Column

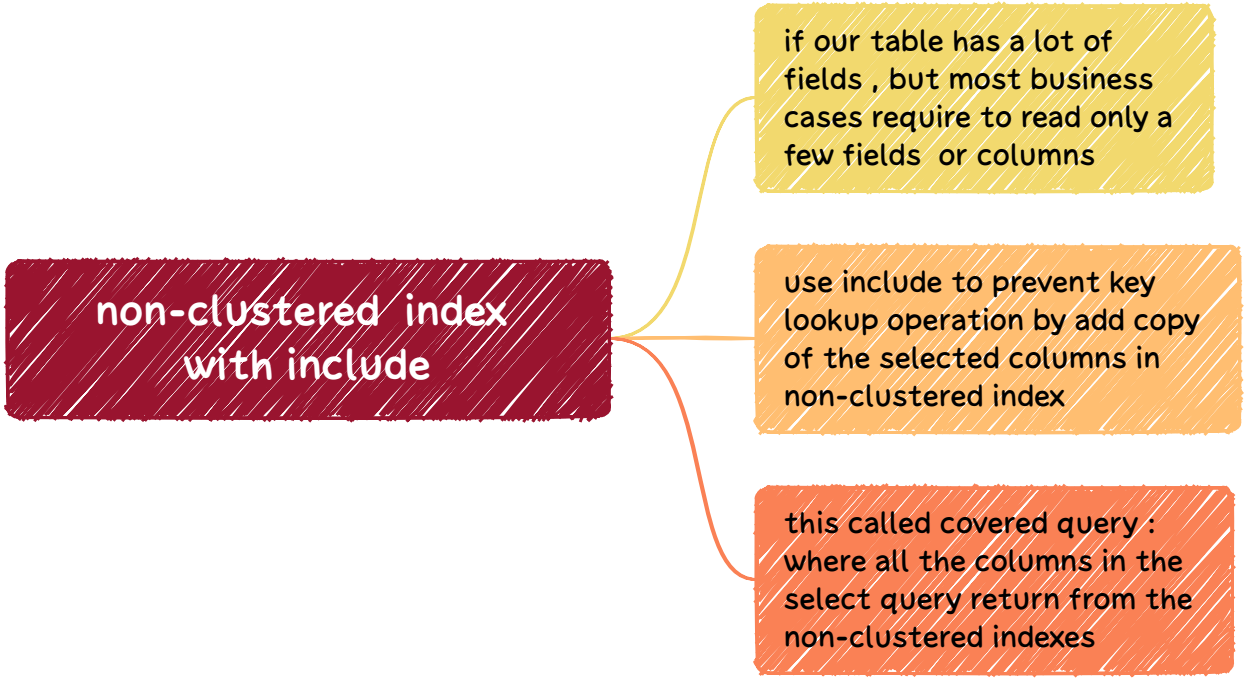


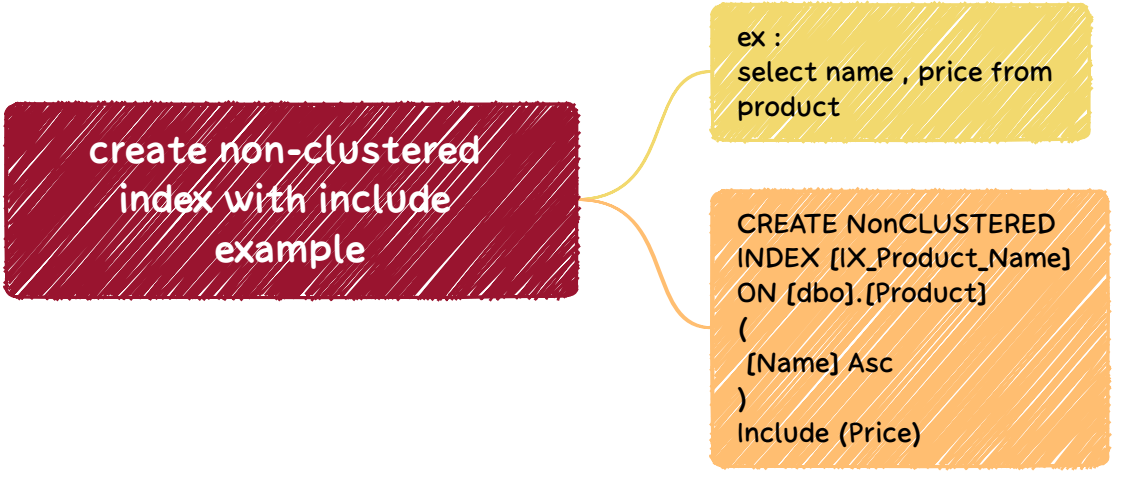
Non Clustered Index On Name Column


SQLQuery1.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (62))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Indexes_Tutorials
Execute
Object Explorer
Connect
DESKTOP-GNFE269\MSSQLSERVER01
Databases
System Databases
Database Snapshots
Indexes_Tutorials
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Product
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - DE..FE269\lenovo (62))*
CREATE NONCLUSTERED INDEX [IX_Product_Name]
ON [dbo].[Product]
(
[Name] Asc
)
SELECT [ID]
,[Name]
,[Price]
,[AvailebleQuantity]
FROM [Indexes_Tutorials].[dbo].[Product]
WHERE NAME = 'nisl Cras'
100 %
Results
Messages
Live Query Statistics
Execution plan
Estimated query progress:100%
Query 1: Query cost (relative to the batch): 100%
SELECT [ID] ,[Name] ,[Price] ,[AvailableQuantity] FROM [Indexes_Tutorials].[dbo].[Product] WHERE NAME = 'nisl Cra
SELECT
Nested Loops
(Inner Join)
32 of
29 (110%)
Index Seek (NonClustered)
[Product].[IX_Product_Name]
32 of
29 (110%)
Key Lookup (Clustered)
[Product].[IX_Product_ID]
32 of
Query executed successfully.
DESKTOP-GNFE269\MSSQLSERVER...
DESKTOP-GNFE269\lenovo...
Indexes_Tutorials
00:00:00
36 rows
Ready
Ln 12
Col 26
Ch 26
INS
24°C
ENG
9:46 PM
9/23/2022


SQLQuery1.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (62))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Indexes_Tutorials
Execute
Object Explorer
Connect
DESKTOP-GNFE269\MSSQLSERVER01
Databases
System Databases
Database Snapshots
Indexes_Tutorials
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Product
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - DE..FE269\lenovo (62))*
CREATE NONCLUSTERED INDEX [IX_Product_Name]
ON [dbo].[Product]
(
[Name] Asc
)
SELECT [ID]
,[Name]
,[Price]
,[AvailebleQuantity]
FROM [Indexes_Tutorials].[dbo].[Product]
WHERE NAME = 'nisl Cras'
100 %
Results
Messages
Live Query Statistics
Execution
Estimated query progress:100%
Query 1: Query cost (relati
SELECT [ID] ,[Name] ,[Price
.[dbo].[Product] WHERE NAME = 'nisl Cra
SELECT
Nested Loops
(Inner Join)
32 of
29 (110%)
Index Se
[Product]
Key Loo
[Product]
Index Seek (NonClustered)
Scan a particular range of rows from a nonclustered index.
Estimated operator progress: 100%
Physical Operation Index Seek
Logical Operation Index Seek
Estimated Execution Mode Row
Storage RowStore
Actual Number of Rows for All Executions 32
Estimated Operator Cost 0.0033135 (4%)
Estimated I/O Cost 0.003125
Estimated Subtree Cost 0.0033135
Estimated CPU Cost 0.0001885
Number of Executions 1
Estimated Number of Executions 1
Estimated Number of Rows Per Execution 28.604
Estimated Number of Rows for All Executions 28.604
Estimated Number of Rows to be Read 28.604
Estimated Row Size 45 B
Ordered True
Node ID 2
Object
[Indexes_Tutorials].[dbo].[Product].[IX_Product_Name]
Output List
Uniq1001, [Indexes_Tutorials].[dbo].[Product].ID, [Indexes_Tutorials].[dbo].[Product].Name
Seek Predicates
Seek Keys[1]: Prefix: [Indexes_Tutorials].[dbo].[Product].Name = Scalar Operator(N'nisl Cras')
Query executed successfully.
TOP-GNFE269\lenovo... Indexes_Tutorials 00:00:00 36 rows
Ready
24°C
ENG
9:47 PM
9/23/2022


SELECT * FROM Product where Name = 'A3'
2- Key Lookup on Clustered Index
Search about Product Has ID = 5
1- Non Clustered Index Seek On Name Column
Get id of Product Has Name = A3
Clustered Index On ID Column
Non Clustered Index On Name Column
First Level : Root Node
page 1000
index key | page id
30 | 1001
50 | 1002
..... | .....
z | 100X
Second Level : intermediate nodes
page 1001
index key | page id
12 | 1003
20 | 1004
..... | .....
page 1002
index key | page id
35 | 1005
45 | 1006
..... | .....
index key | page id
.... | .....
.... | .....
Z | .....
page 1003
index key | page id
3 | 7001
6 | 7002
..... | .....
page 1004
index key | page id
..... | .....
..... | .....
..... | .....
Third Level : Leaf Nodes
page 7001
ID | name | price | Qty
1 | C2 | ... | ...
2 | D1 | ... | ...
3 | A2 | ... | ...
page 7002
ID | name | price | Qty
4 | B1 | ... | ...
5 | A3 | ... | ...
6 | C1 | ... | ...
page 7003
ID | name | price | Qty
7 | D2 | ... | ...
8 | B2 | ... | ...
9 | A1 | ... | ...
First Level : Root Node
page 1000
index key | page id
A | 2001
D | 2002
..... | .....
z | 200X
Second Level : intermediate nodes
page 2001
index key | page id
A | 9003
B | 9003
C | 9004
page 2002
index key | page id
D | 9004
E | 9005
..... | .....
index key | page id
.... | .....
.... | .....
Z | .....
Third Level : Leaf Nodes
page 9002
Non-Clustered Index (name) | Clustered index (id)
A1 | 9
A2 | 3
A3 | 5
B1 | 4
B2 | 8
page 9002
Non-Clustered Index (name) | Clustered index (id)
C1 | 6
C2 | 1
D1 | 2
D2 | 7

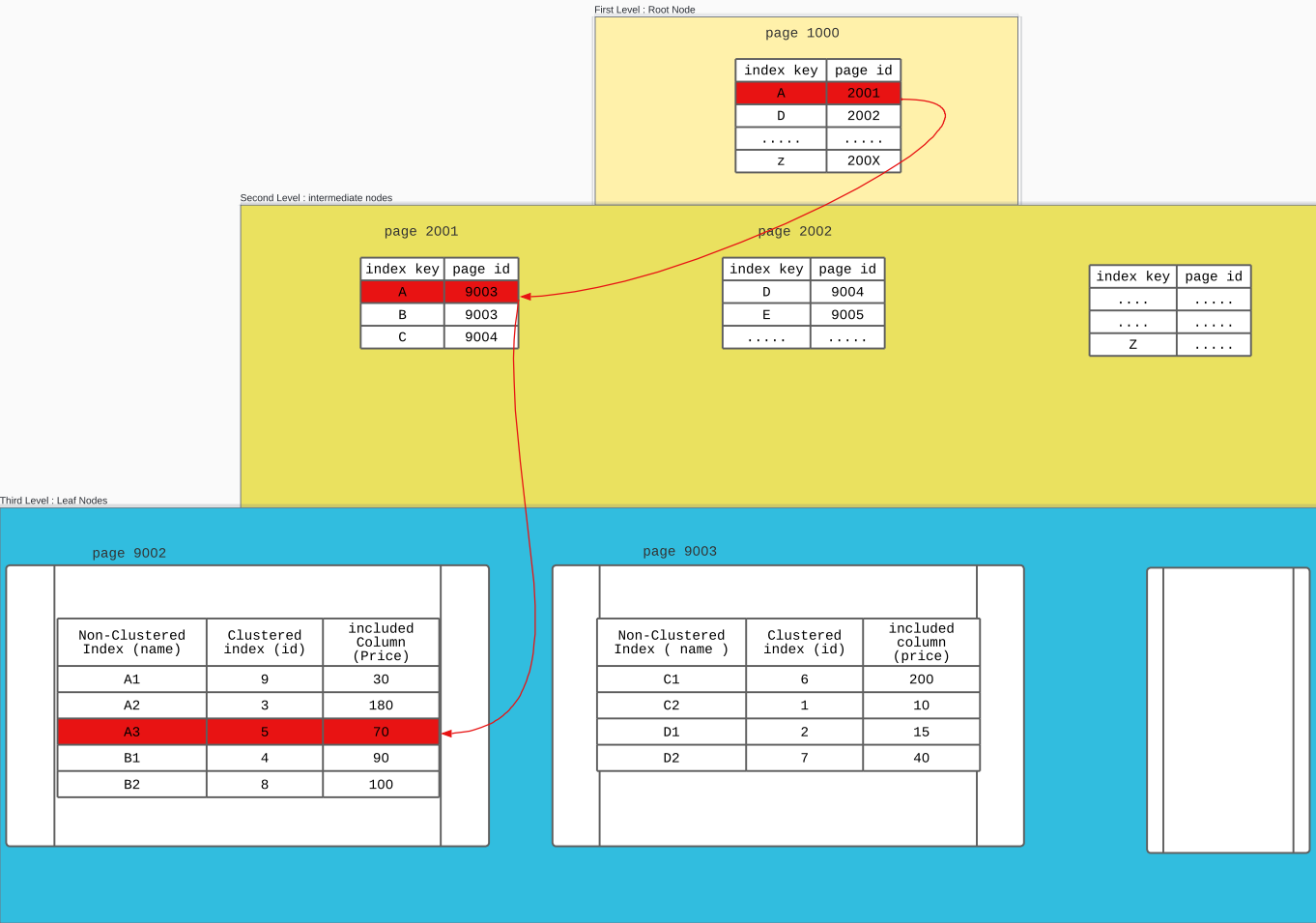


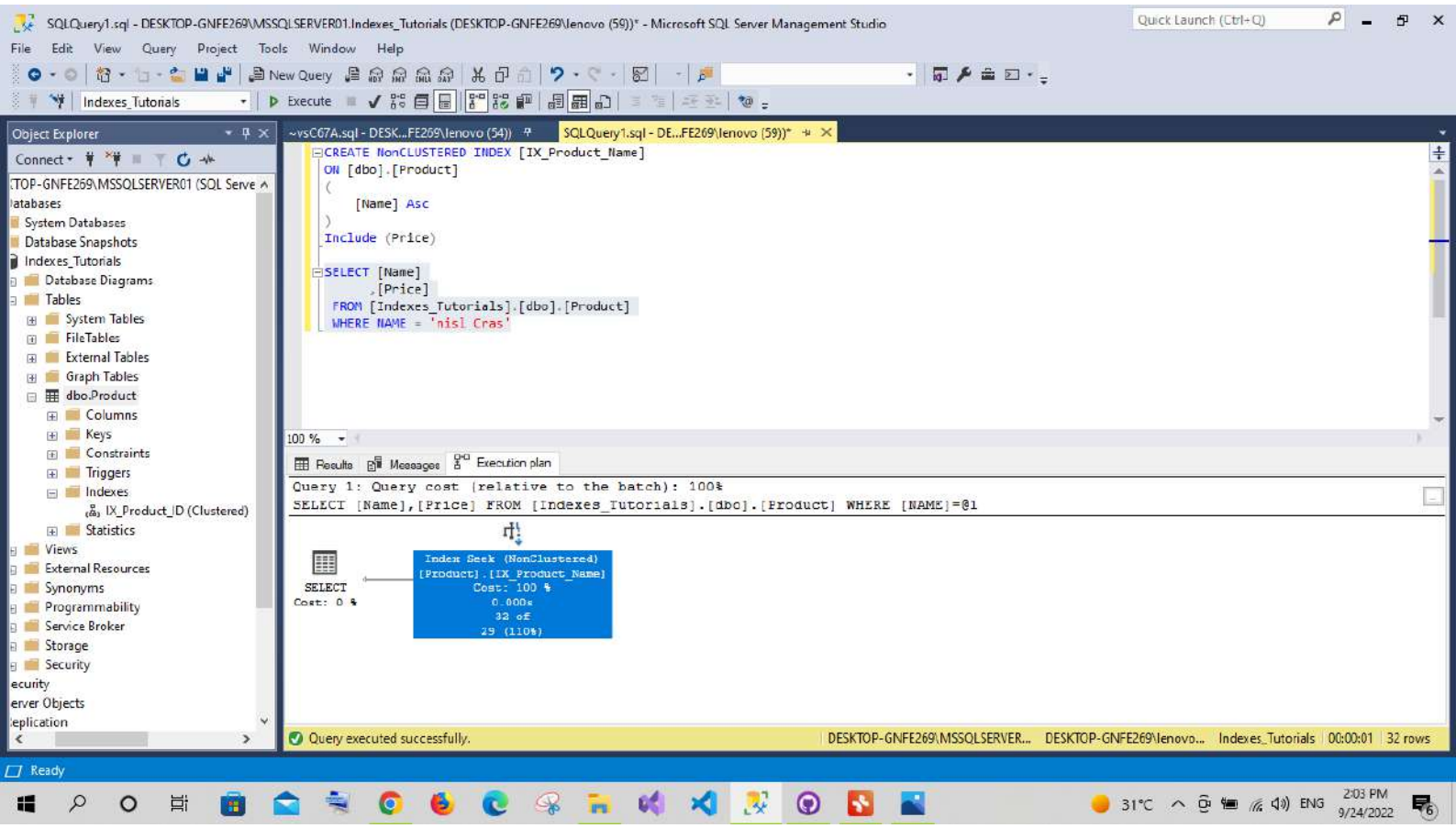


```
SELECT NAME , PRICE FROM Product
Where Name = 'A3'
```

1- Non Clustered Index Seek

Non Clustered Index On Name Column with include (Price Column)




SQLQuery1.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (59))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Indexes_Tutorials
Execute
Object Explorer
Connect
TOP-GNFE269\MSSQLSERVER01 (SQL Serve
latabases
System Databases
Database Snapshots
Indexes_Tutorials
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Product
Columns
Keys
Constraints
Triggers
Indexes
IX_Product_ID (Clustered)
Statistics
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
ecurity
erver Objects
eplication
~vsC67A.sql - DESK...FE269\lenovo (54))
SQLQuery1.sql - DE...FE269\lenovo (59))*
CREATE NonCLUSTERED INDEX [IX_Product_Name]
ON [dbo].[Product]
(
[Name] Asc
)
Include (Price)
SELECT [Name]
,[Price]
FROM [Indexes_Tutorials].[dbo].[Product]
WHERE NAME = 'nisl Cras'
100 %
Results
Messages
Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT [Name],[Price] FROM [Indexes_Tutorials].[dbo].[Product] WHERE [NAME]=@1
SELECT
Cost: 0 %
Index Seek (NonClustered)
[Product].[IX_Product_Name]
Cost: 100 %
0.000s
32 of
29 (110%)
Query executed successfully.
DESKTOP-GNFE269\MSSQLSERVER...
DESKTOP-GNFE269\lenovo...
Indexes_Tutorials
00:00:01
32 rows
Ready
31°C
ENG
2:03 PM
9/24/2022