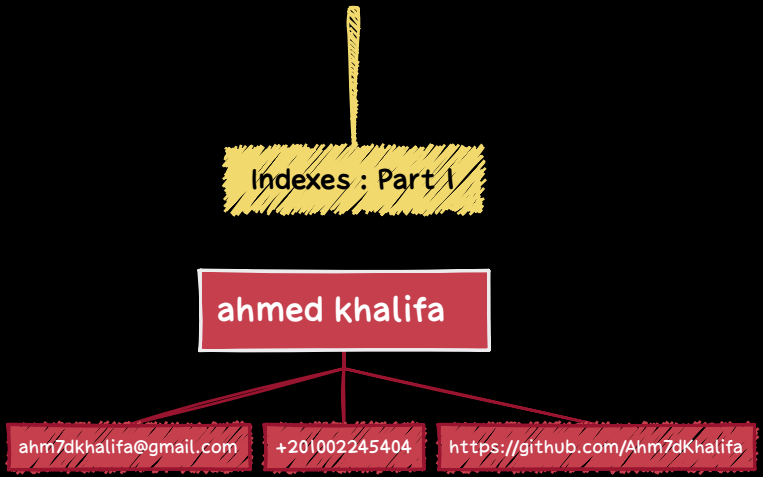
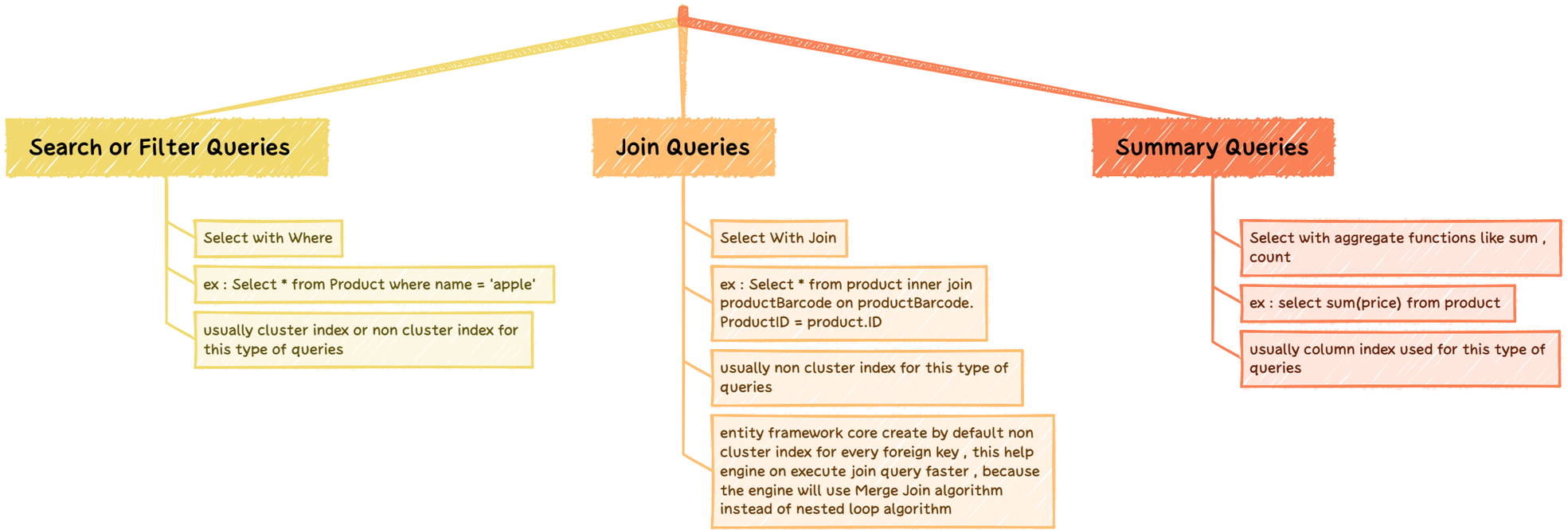
Physical Database Design



What is index ?

- Data structure associated with a table or view to help database engine to retrieve data very fast
- Type of data structure :
  - usually but not always the data structure is B-tree
  - the type of data structure is depend on some factors like : type of index , type of sql provider like : oracle , sql server or mysql ... and so on

Types Of SQL Queries That Can Use Indexes :



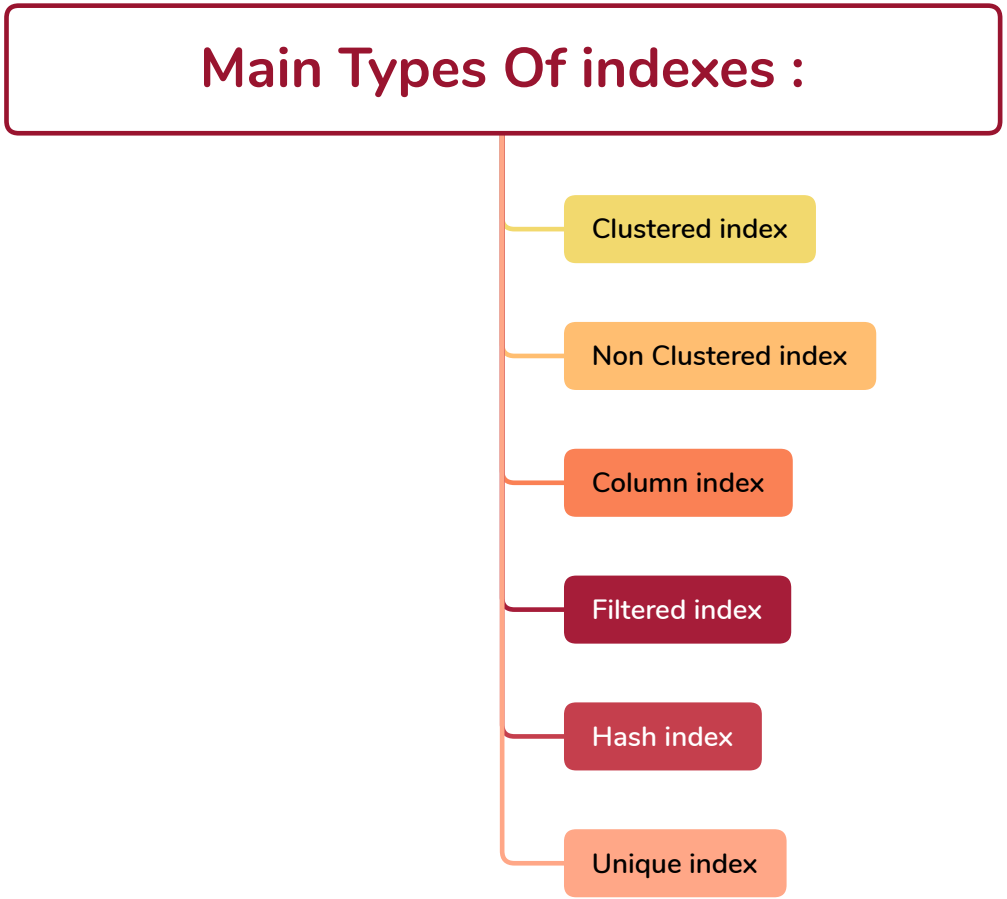
Advantages Of indexes :)

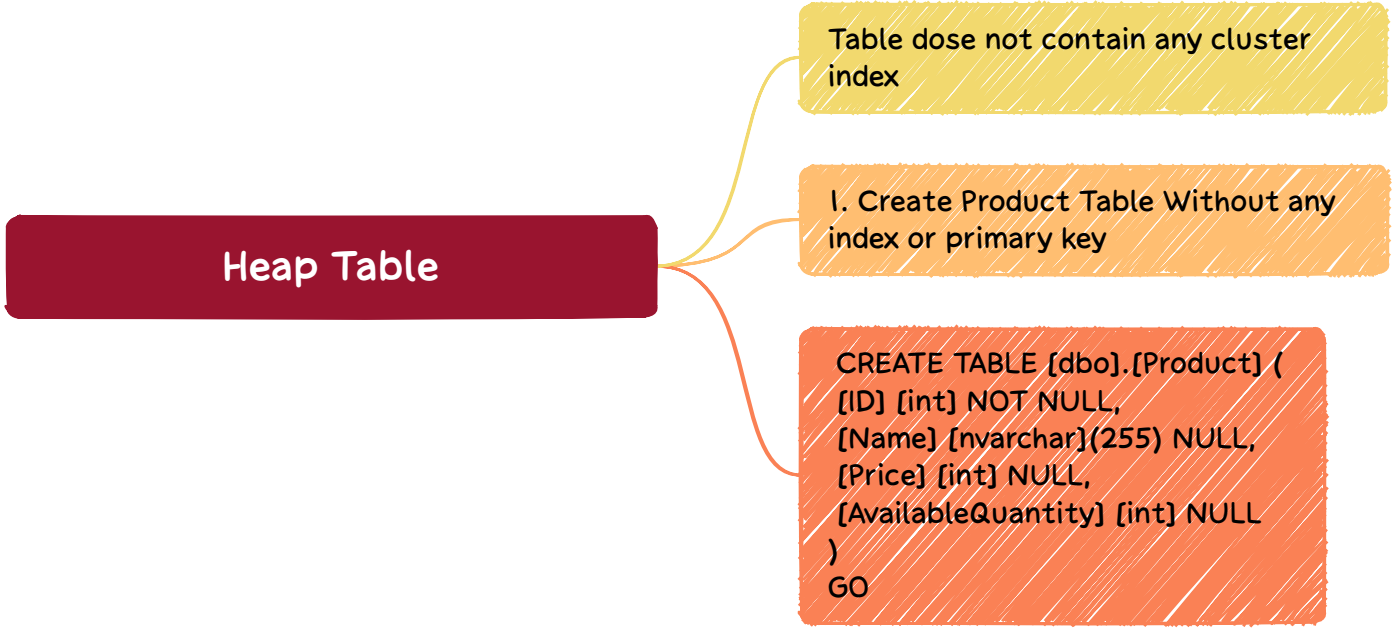
- Help database engine to retrieve data or execute select queries very fast with high performance
- Help database engine to load minimum number of pages or rows from disk to memory
- save database server memory or buffers space as possible

Disadvantages of indexes :(

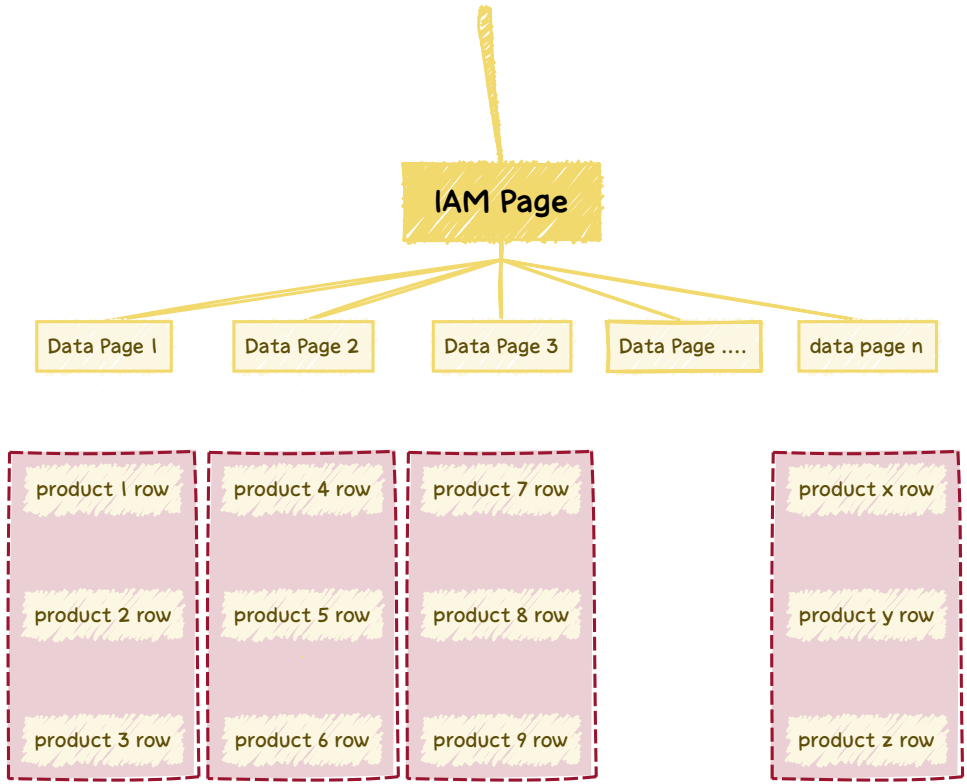
- indexes increase performance of read queries but when there a lot of indexes this will be decrease performance of write queries like : insert , update , delete.
- indexes need extra space
- the two previous problems can be solved if you design your index correctly and understand your business and needs

Main Types Of indexes :




Heap Table
Table dose not contain any cluster index
1. Create Product Table Without any index or primary key
CREATE TABLE [dbo].[Product] (
[ID] [int] NOT NULL,
[Name] [nvarchar](255) NULL,
[Price] [int] NULL,
[AvailableQuantity] [int] NULL
)
GO

Product Heap Table Physical Representation



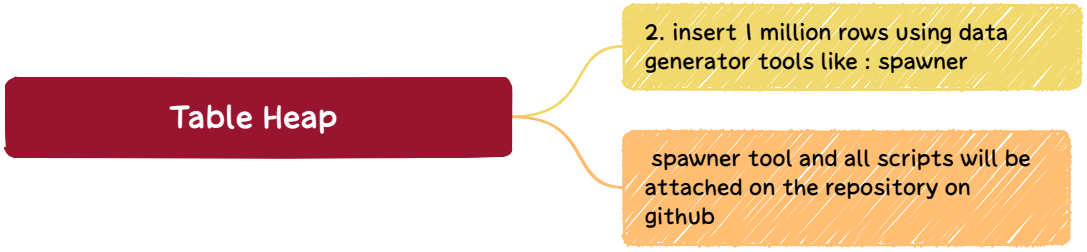


Table Heap
2. insert 1 million rows using data generator tools like : spawner
spawner tool and all scripts will be attached on the repository on github

Product				
ID	Name	Price	Available Quantity	Page ID
1	Product 1	20	5	1
2	Product 2	10	2	1
3	Product 3	40	6	1
4	Product 4	70	3	2
5	Product 5	5	7	2
6	Product 6	120	1	2
....				
1,000,000	Product 1,000,000	15	8	N

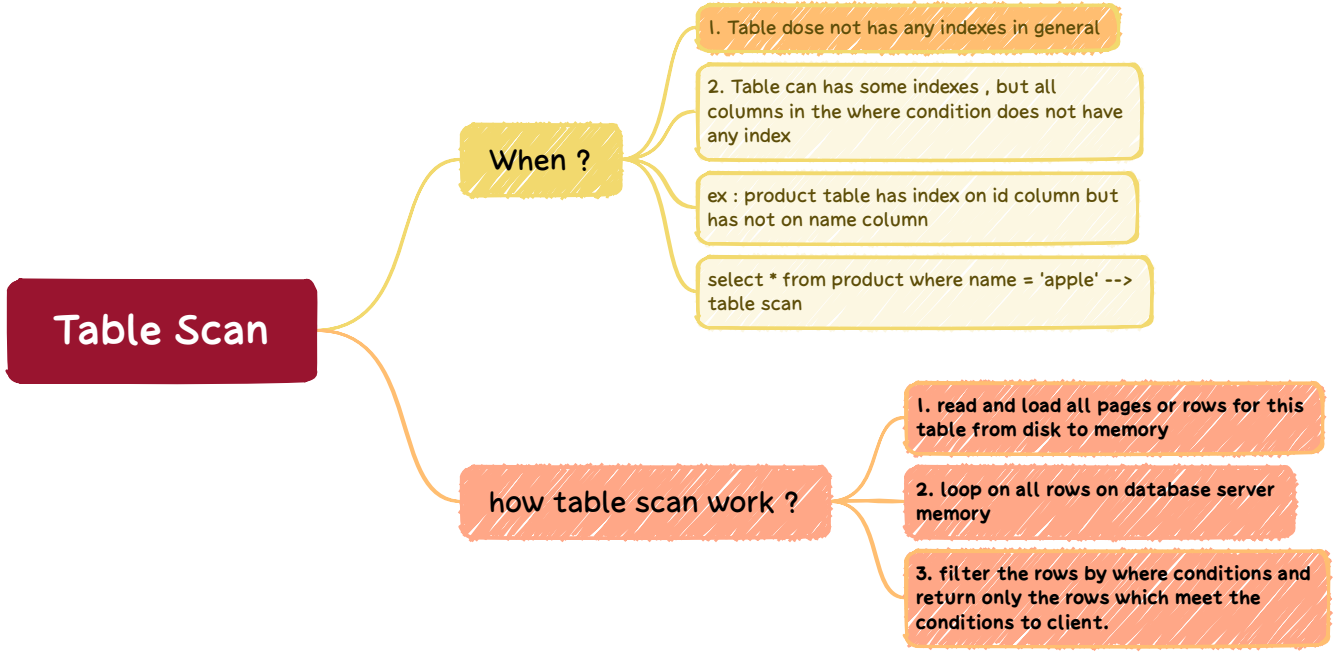


Table Scan
When ?
1. Table dose not has any indexes in general
2. Table can has some indexes , but all columns in the where condition does not have any index
ex : product table has index on id column but has not on name column
select * from product where name = 'apple' --> table scan
how table scan work ?
1. read and load all pages or rows for this table from disk to memory
2. loop on all rows on database server memory
3. filter the rows by where conditions and return only the rows which meet the conditions to client.

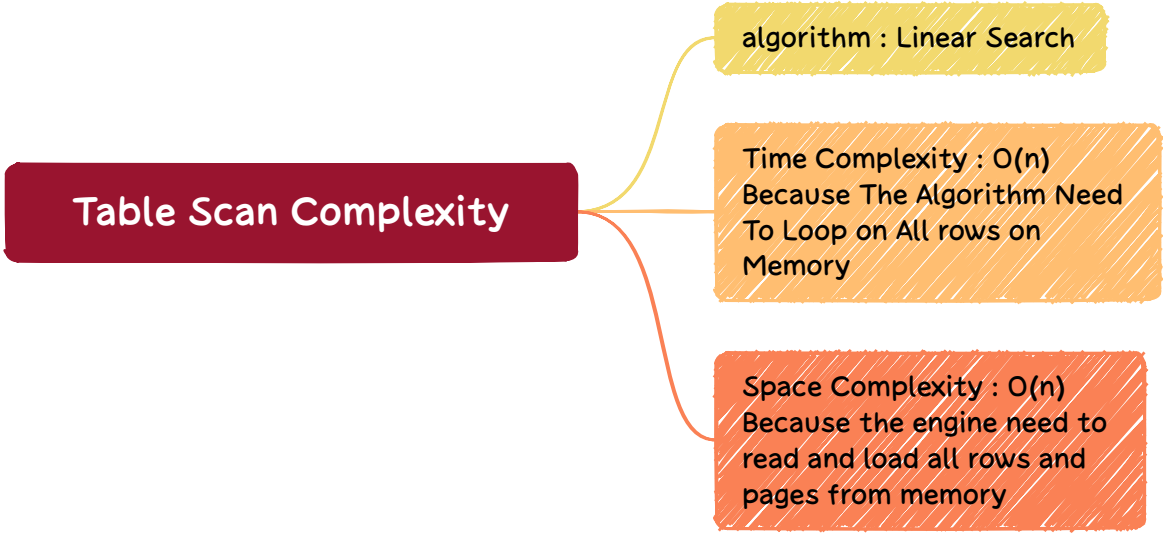


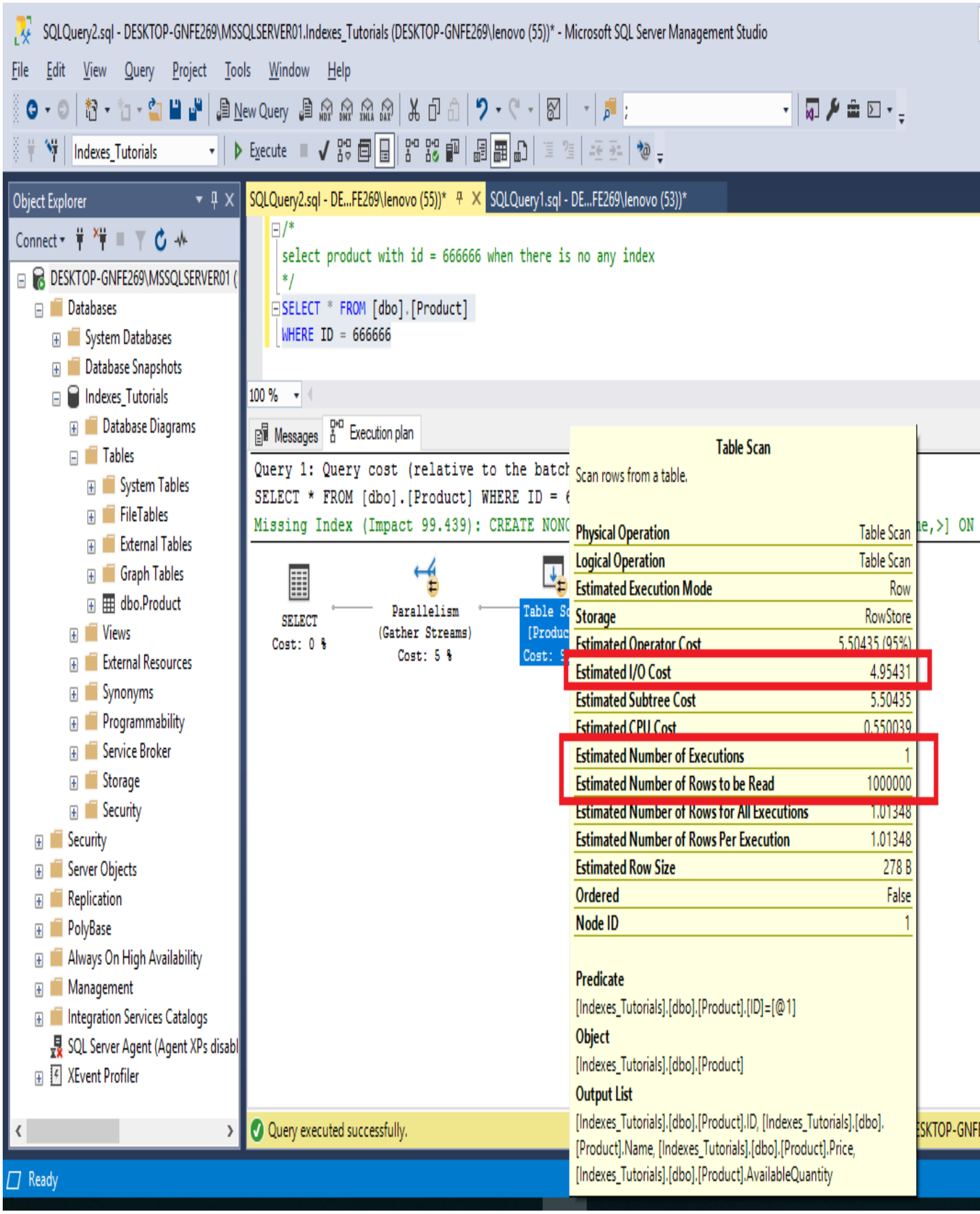
Table Scan Complexity
algorithm : Linear Search
Time Complexity : O(n)
Because The Algorithm Need To Loop on All rows on Memory
Space Complexity : O(n)
Because the engine need to read and load all rows and pages from memory

Heap Table

3. Run this Query and Open Estimated Execution Plan

SELECT * FROM [dbo].[Product] WHERE ID = 666666

The Result as you see on The Next Image , Read and Load All Rows To Return Only One Row


SQLQuery2.sql - DESKTOP-GNFE269\MSSQLSERVER01.Indexes_Tutorials (DESKTOP-GNFE269\lenovo (55))* - Microsoft SQL Server Management Studio
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
SQLQuery2.sql - DE...FE269\lenovo (55))*
SQLQuery1.sql - DE...FE269\lenovo (53))*
/*
select product with id = 666666 when there is no any index
*/
SELECT * FROM [dbo].[Product]
WHERE ID = 666666
100 %
Messages
Execution plan
Query 1: Query cost (relative to the batch
SELECT * FROM [dbo].[Product] WHERE ID = 6
Missing Index (Impact 99.439): CREATE NONC
SELECT
Cost: 0 %
Parallelism
(Gather Streams)
Cost: 5 %
Table Sc
[Product
Cost: 9
Table Scan
Scan rows from a table.
Physical Operation Table Scan
Logical Operation Table Scan
Estimated Execution Mode Row
Storage RowStore
Estimated Operator Cost 5.50435 (95%)
Estimated I/O Cost 4.95431
Estimated Subtree Cost 5.50435
Estimated CPU Cost 0.550039
Estimated Number of Executions 1
Estimated Number of Rows to be Read 1000000
Estimated Number of Rows for All Executions 1.01348
Estimated Number of Rows Per Execution 1.01348
Estimated Row Size 278 B
Ordered False
Node ID 1
Predicate
[Indexes_Tutorials].[dbo].[Product].[ID]=[@1]
Object
[Indexes_Tutorials].[dbo].[Product]
Output List
[Indexes_Tutorials].[dbo].[Product].ID, [Indexes_Tutorials].[dbo].[Product].Name, [Indexes_Tutorials].[dbo].[Product].Price, [Indexes_Tutorials].[dbo].[Product].AvailableQuantity
Query executed successfully.
Ready