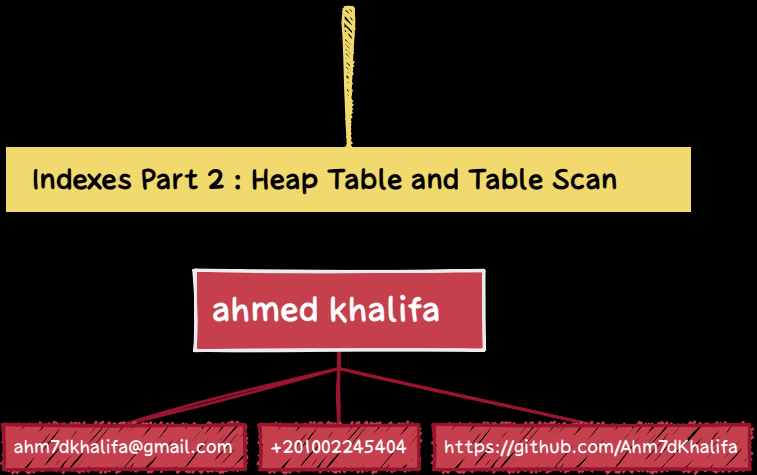
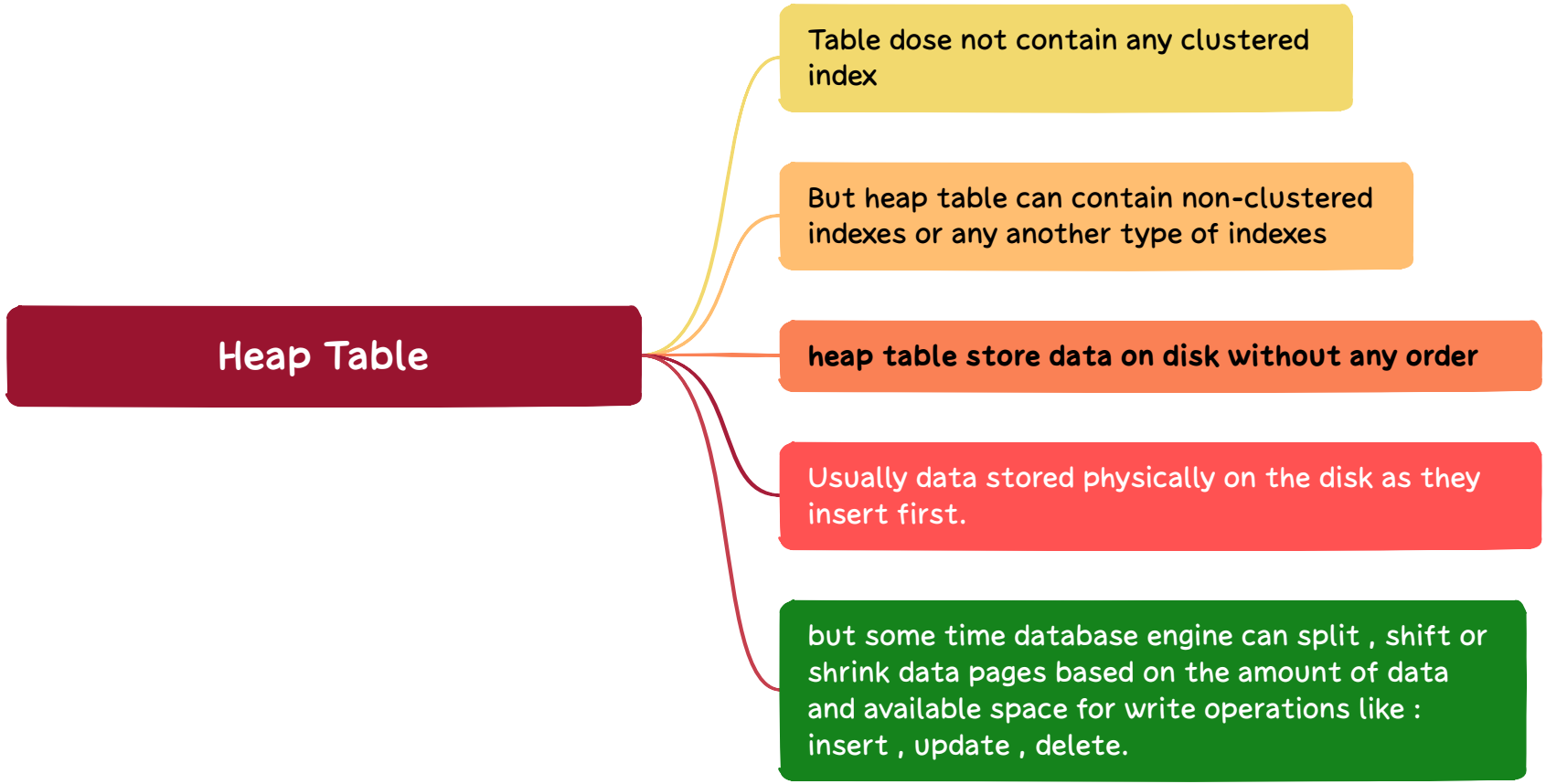
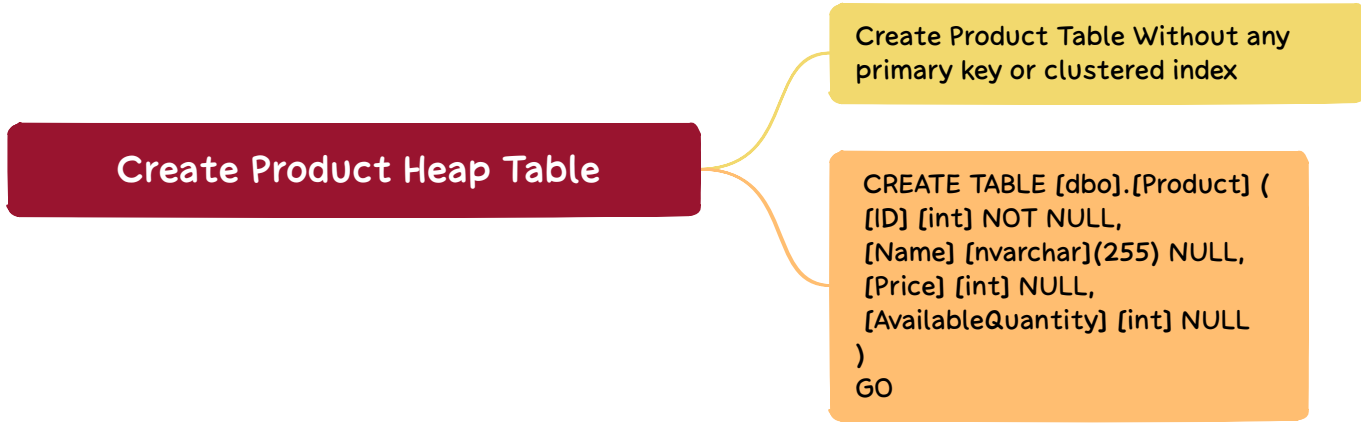
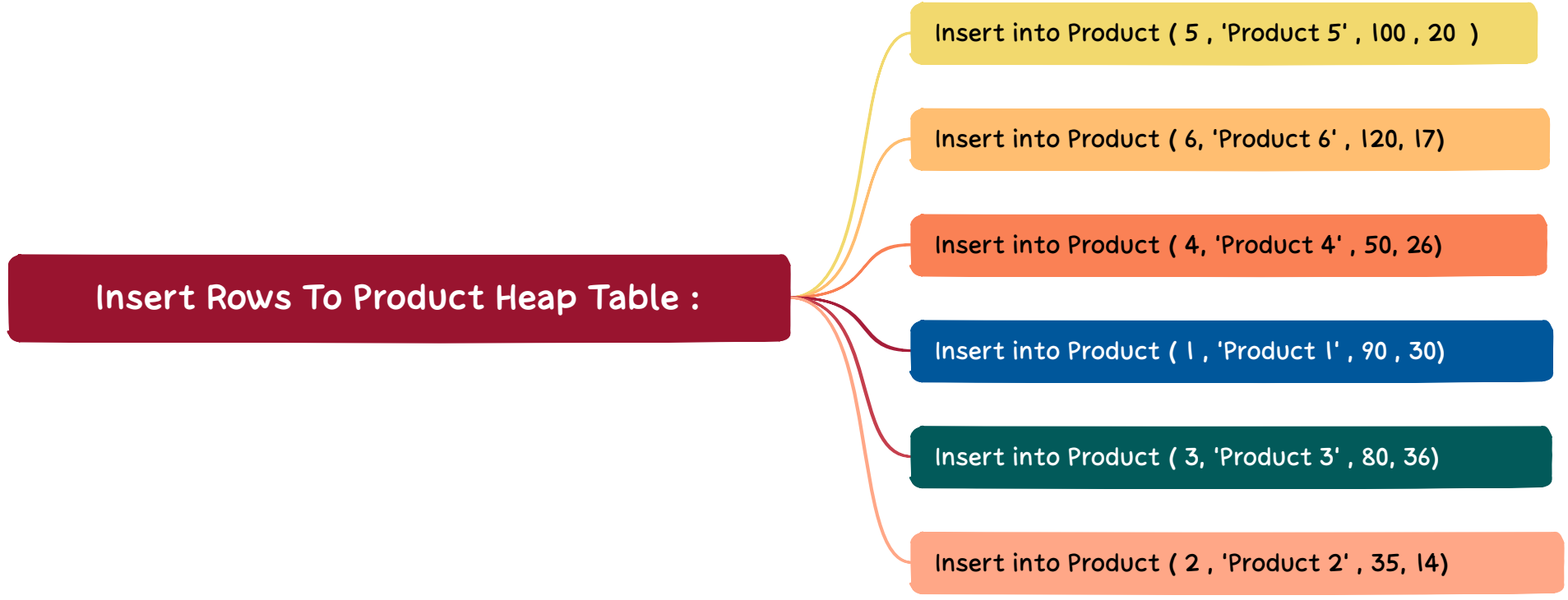
Physical Database Design

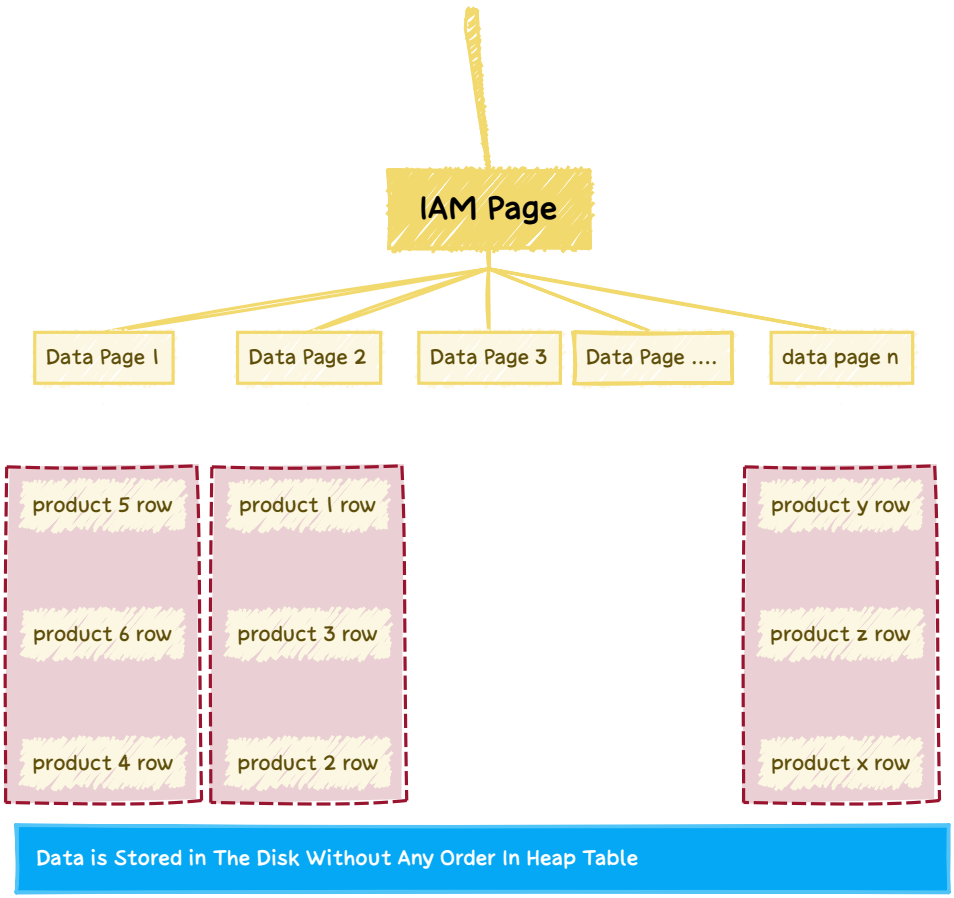








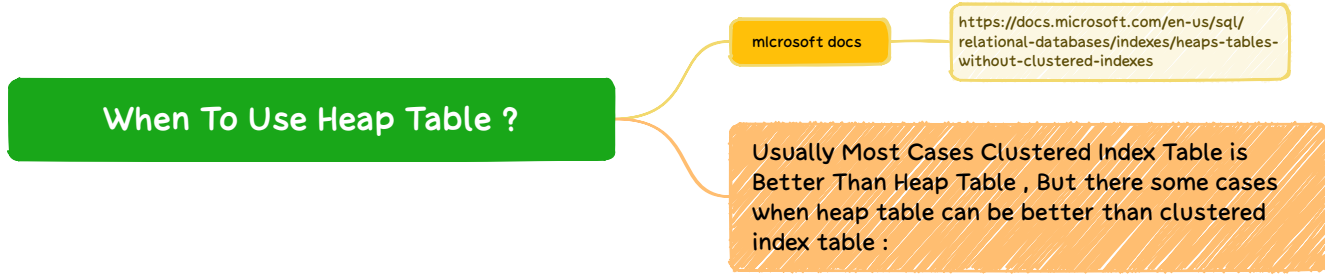
Product Heap Table Physical Representation On Disk

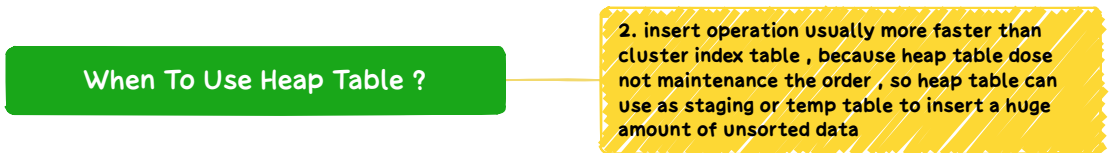


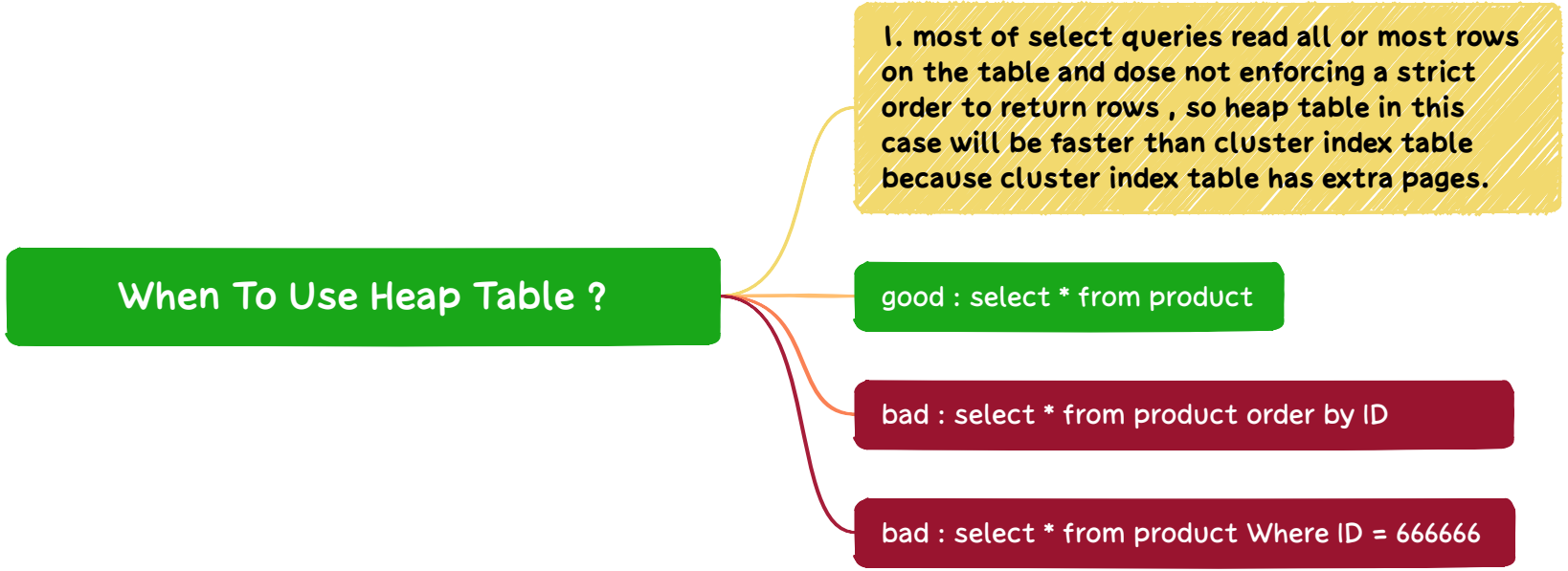
Select * From Product

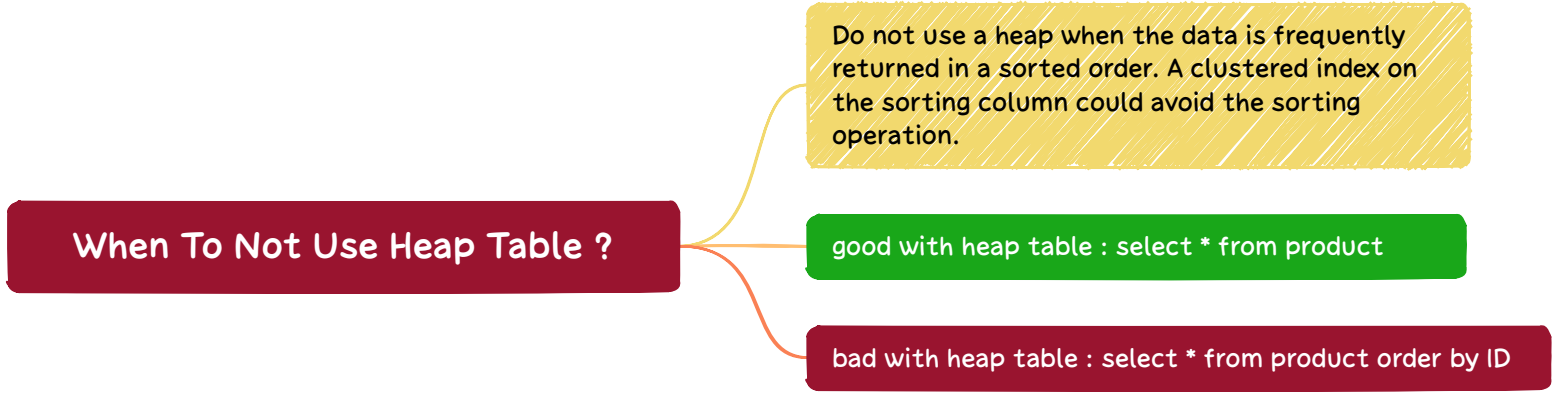
Product Heap Table Logical Representation On Memory

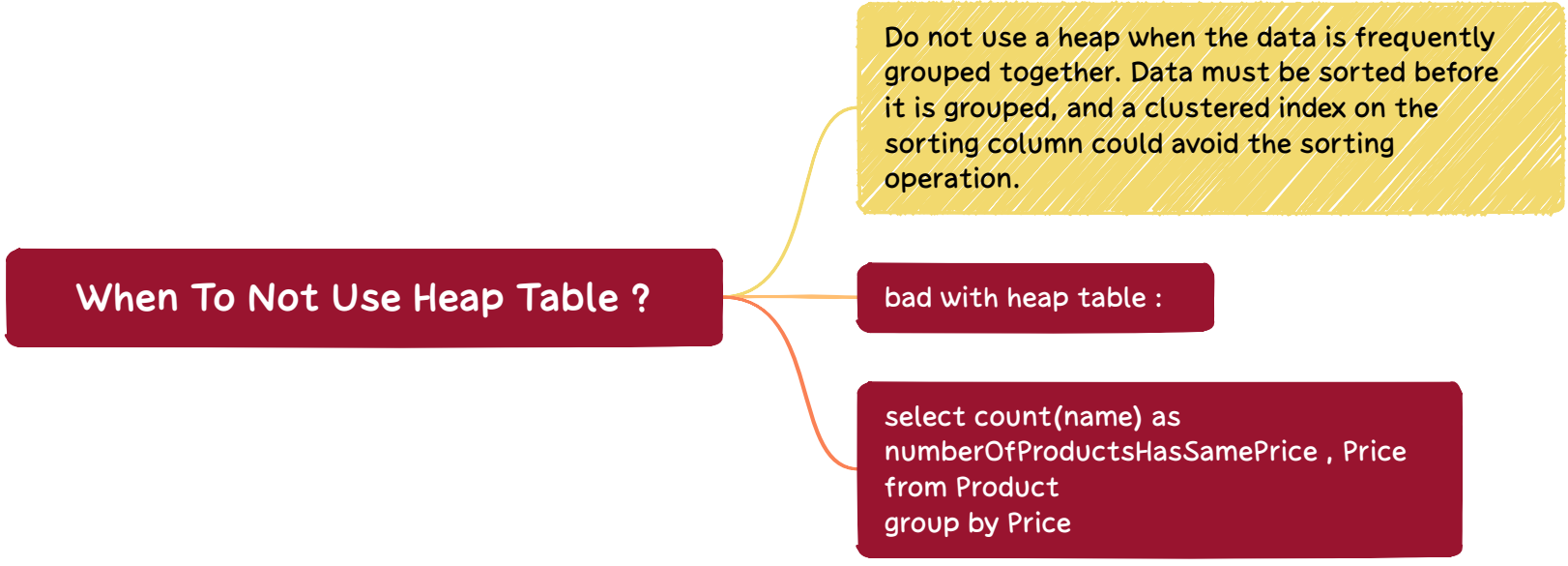
	ID	Name	Price	Available Quantity
1	5	product 5	100	20
2	6	product 6	120	17
3	4	product 4	50	26
4	1	product 1	90	30
5	3	product 3	80	36
6	2	product 2	35	14
.....	....	.....	.....	.....

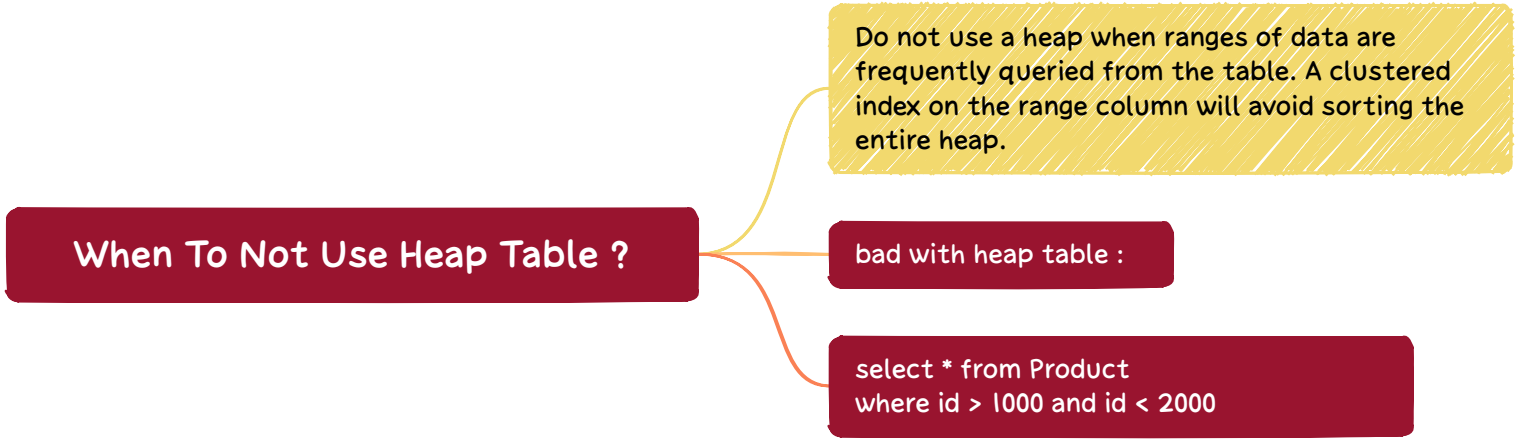










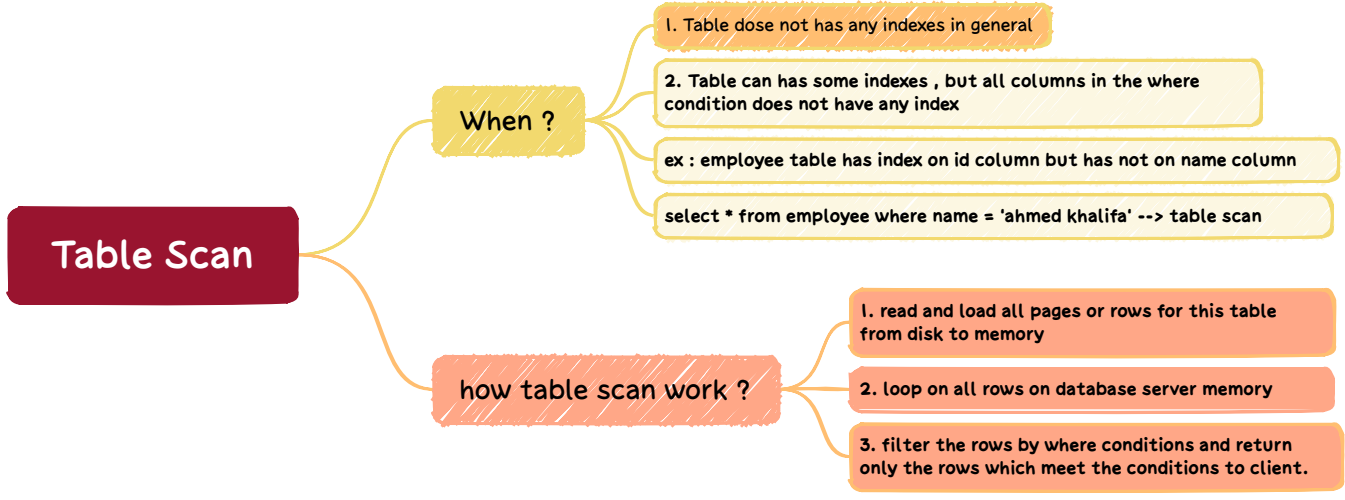


When To Not Use Heap Table ?

Do not use a heap when there are no nonclustered indexes and the table is large, unless you intend to return the entire table content without any specified order. In a heap, all rows of the heap must be read to find any row

When To Not Use Heap Table ?

Do not use a heap if the data is frequently updated. If you update a record and the update uses more space in the data pages than they are currently using, the record has to be moved to a data page that has enough free space. This creates a forwarded record pointing to the new location of the data, and forwarding pointer has to be written in the page that held that data previously, to indicate the new physical location. This introduces fragmentation in the heap. When scanning a heap, these pointers must be followed which limits read-ahead performance, and can incur additional I/O which reduces scan performance.



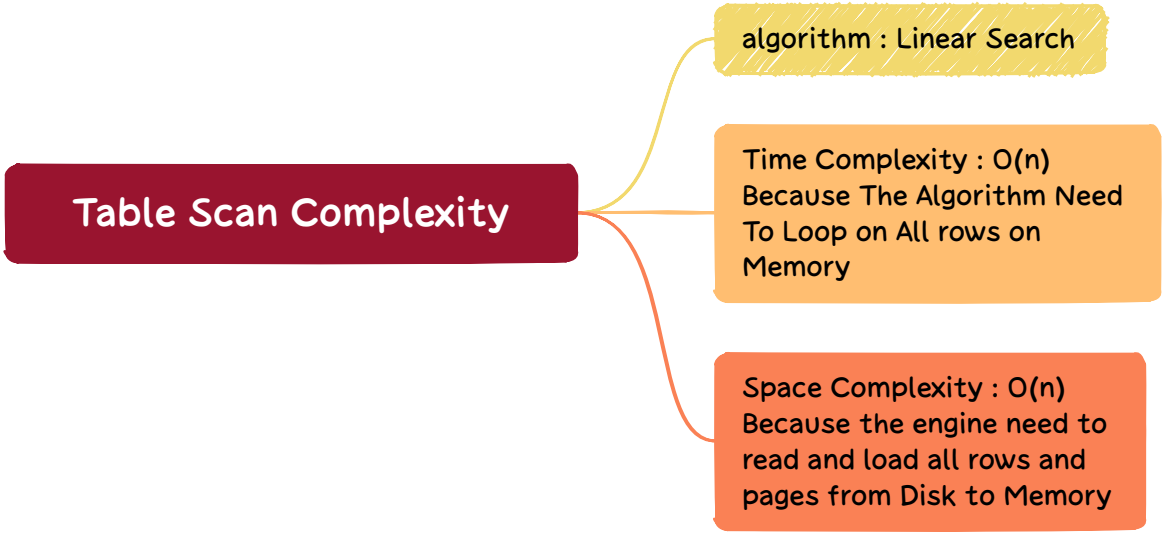
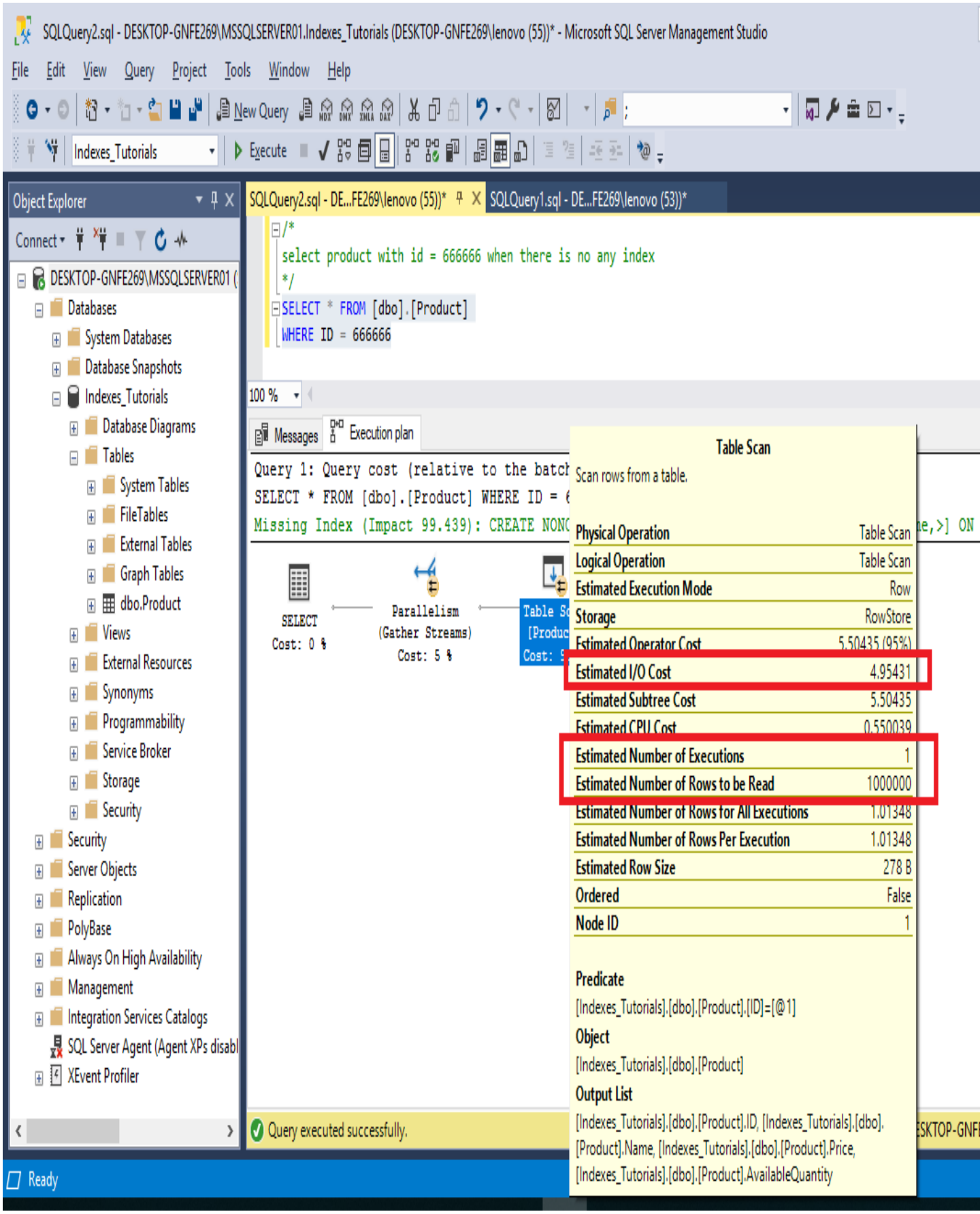


Table Scan With one Million Rows Example

- Insert 1 Million Rows By Data Generator Tools Like Spawner
- all Tools and Scripts are attached on github repository : https://github.com/Ahm7dKhalifa/Database-Design
- Note To Remember : Product Table is Heap , and Dose Not Has Cluster Index
- 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