

**Cairo University**



**Faculty Of Engineering**

Advanced Database System

Report

Ahmed Mohamed Ismail Nabeel 1180501

Mostafa Ashraf Ahmed 1180406

Moaz Mohamed Elsherbini 1180528

Nader Youhanna Adib 1180477

Table of Contents

[System Information 4](#_Toc123753102)

[Query 1 5](#_Toc123753103)

[Before Optimization 5](#_Toc123753104)

[Execution Plan 5](#_Toc123753105)

[After Optimization 5](#_Toc123753106)

[Execution Plan 6](#_Toc123753107)

[Optimization Done 6](#_Toc123753108)

[Theoretical Parallel Query Processing Report 6](#_Toc123753109)

[Query 2 7](#_Toc123753110)

[Before Optimization 7](#_Toc123753111)

[Execution Plan 7](#_Toc123753112)

[After Optimization 7](#_Toc123753113)

[Execution plan 8](#_Toc123753114)

[Optimization Done 8](#_Toc123753115)

[Theoretical Parallel Query Processing Report 8](#_Toc123753116)

[Query 3 9](#_Toc123753117)

[Before Optimization 9](#_Toc123753118)

[Execution Plan 9](#_Toc123753119)

[After Optimization 9](#_Toc123753120)

[Stored Procedure 10](#_Toc123753121)

[Execution Plan 10](#_Toc123753122)

[Optimization Done 10](#_Toc123753123)

[Theoretical Parallel Query Processing Report 10](#_Toc123753124)

[Database Statistics 11](#_Toc123753125)

[Times for Different Database Sizes 12](#_Toc123753126)

[Time Analysis 13](#_Toc123753127)

[NoSQL Queries 15](#_Toc123753128)

[Query 1 15](#_Toc123753129)

[Query 2 16](#_Toc123753130)

[Query 3 18](#_Toc123753131)

[Optimizations 19](#_Toc123753132)

[Schema Enhancement 19](#_Toc123753133)

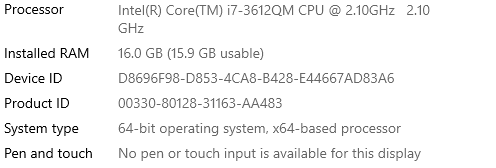
[Memory and Cache Management Enhancement 19](#_Toc123753134)

[Indexes Modifications 19](#_Toc123753135)

[Query Rewriting Modifications 19](#_Toc123753136)

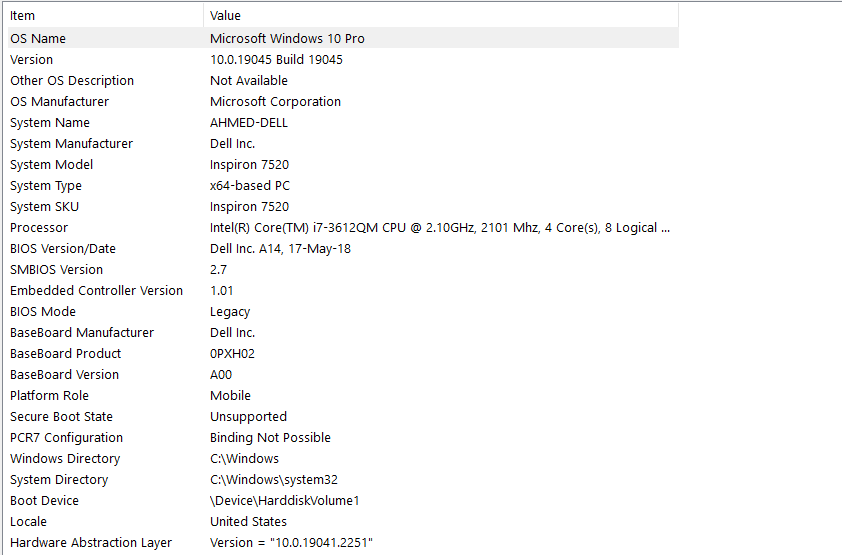
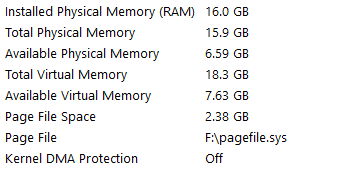
[Conclusion 19](#_Toc123753137)

# System Information



Operating System: Windows 10 Pro

Hard disk: 1 TB SSD

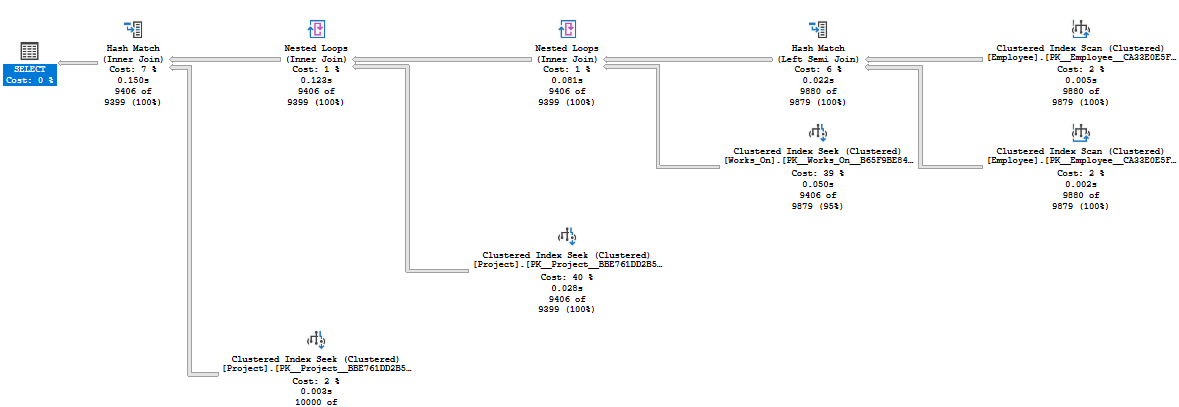


# Query 1

## Before Optimization



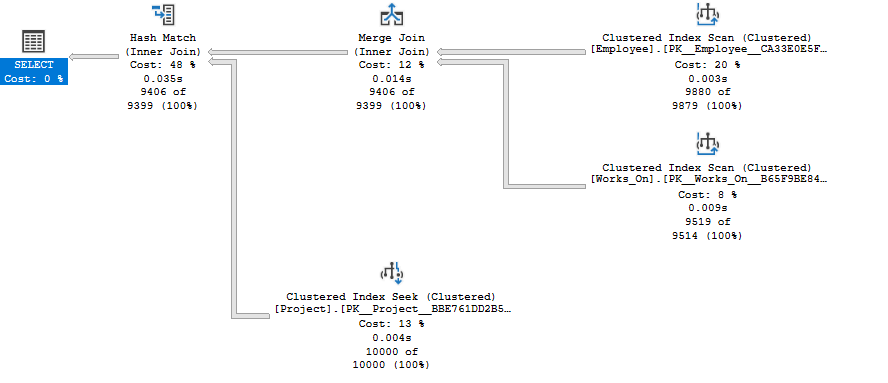
### Execution Plan



## After Optimization



### Execution Plan



## Optimization Done

* INNER JOIN instead of NESTED LOOP JOIN
* Non-clustered index on Ssn & Pnumber

|  |  |  |
| --- | --- | --- |
| Action | CPU Time (s) | Elapsed Time (s) |
| Query 1 (No optimization) | 0.688 | 2.180 |
| Query 1 (Optimized INNER JOIN) | 0.136 | 2.106 |
| Adding non-clustering index | 0.123 | 1.902 |
| NoSQL MongoDB | 2.223 |  |

## Theoretical Parallel Query Processing Report

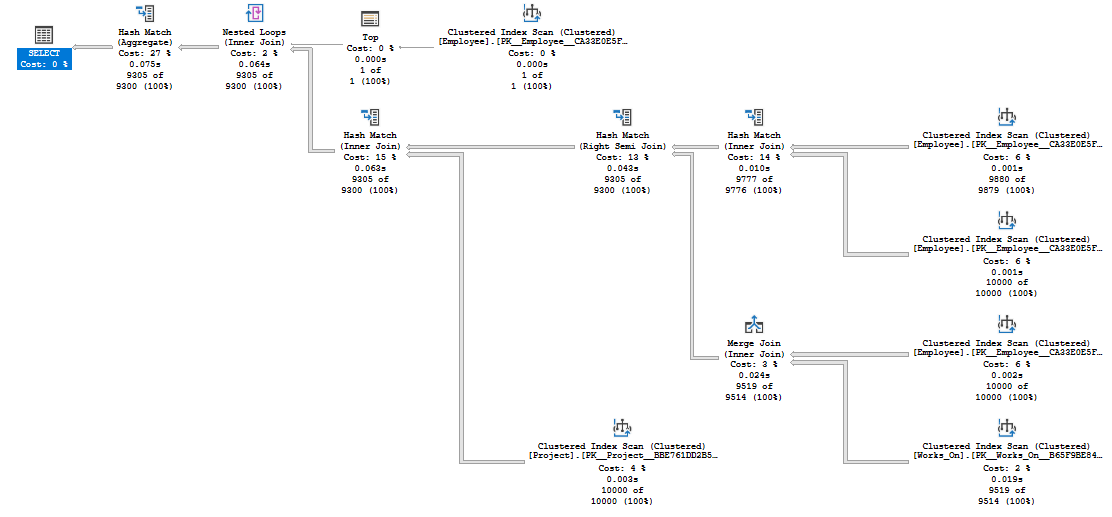
* Clustered index scan on Employee table
* Clustered index scan on Project table
* Clustered index scan on Works\_On table

# Query 2

## Before Optimization



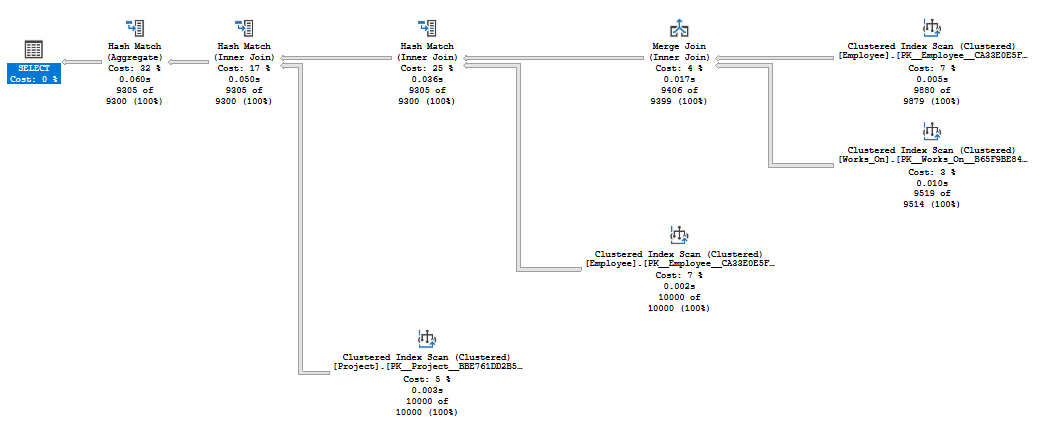
### Execution Plan



## After Optimization



### Execution plan



## Optimization Done

* INNER JOIN instead of nested loop.
* Non-clustered index on Ssn, Pnumber, Salary and Hours.

|  |  |  |
| --- | --- | --- |
| Action | CPU Time (s) | Elapsed Time (s) |
| Query 1 (No optimization) | 0.476 | 4.467 |
| Query 1 (Optimized INNER JOIN) | 0.438 | 1.922 |
| Adding non-clustering index | 0.317 | 1.800 |
| NoSQL MongoDB | 310.751 |  |

## Theoretical Parallel Query Processing Report

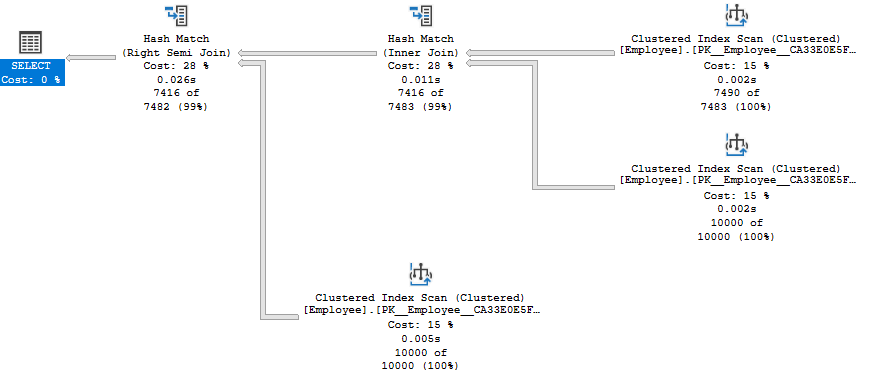
* Clustered index scan on Project table
* Clustered index scan on Works\_On table
* Clustered index scan on Employee

# Query 3

## Before Optimization



### Execution Plan



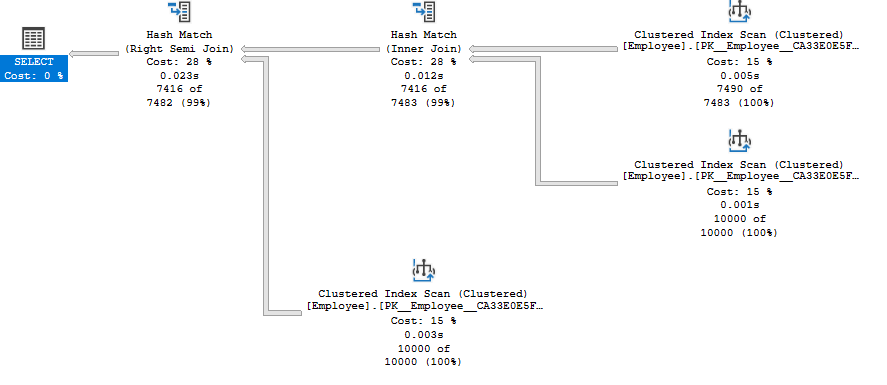
## After Optimization



### Stored Procedure



### Execution Plan



## Optimization Done

* Used stored procedure.
* Non-clustered index on Ssn & Bdate.

|  |  |  |
| --- | --- | --- |
| Action | CPU Time (s) | Elapsed Time (s) |
| Query 1 (No optimization) | 0.139 | 1.983 |
| Query 1 (Stored Procedure) | 0.145 | 1.751 |
| Adding non-clustering index | 0.121 | 2.002 |
| NoSQL MongoDB | 17.946 |  |

## Theoretical Parallel Query Processing Report

* Clustered index scan on Employee

# Database Statistics

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name | Row Count | Main Key | Indexes | FK | Identity Column | Max Row Size (Bytes) |
| Employee | 1M | Yes | 1 | 2 | Yes | 71 |
| Department | 1M | Yes | 1 | 1 | Yes | 19 |
| Project | 1M | Yes | 1 | 1 | Yes | 38 |
| Works\_On | 1M | No | 1 | 2 | No | 12 |
| Department\_Location | 1M | No | 1 | 1 | No | 19 |
| Dependent | 1M | No | 1 | 1 | No | 38 |

# Times for Different Database Sizes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Query Size | 10K (s) | 100K (s) | 1M (s) | 10M (s) |
| Query 1 (Non-optimized)  CPU Time | 0.688 | 2.183 | 19.771 | 184.889 |
| Query 1 (Non-optimized)  Elapsed Time | 2.180 | 4.666 | 33.409 | 290.898 |
| Query 1 (Optimized)  CPU Time | 0.123 | 0.769 | 7.126 | 62.678 |
| Query 1 (Optimized)  Elapsed Time | 1.902 | 3.315 | 20.760 | 170.614 |
| Query 2 (Non-optimized)  CPU Time | 0.476 | 1.739 | 12.106 | 120.700 |
| Query 2 (Non-optimized)  Elapsed Time | 4.467 | 4.348 | 25.066 | 230.224 |
| Query 2 (Optimized)  CPU Time | 0.317 | 0.997 | 10.092 | 101.365 |
| Query 2 (Optimized)  Elapsed Time | 1.800 | 3.495 | 22.735 | 209.833 |
| Query 3 (Non-optimized)  CPU Time | 0.139 | 0.944 | 7.204 | 76.916 |
| Query 3 (Non-optimized)  Elapsed Time | 1.983 | 3.432 | 19.330 | 187.511 |
| Query 3 (Optimized)  CPU Time | 0.121 | 0.976 | 7.184 | 73.292 |
| Query 3 (Optimized)  Elapsed Time | 2.002 | 3.542 | 19.047 | 179.315 |

# Time Analysis

# NoSQL Queries

## Query 1



## Query 2





## Query 3



# Optimizations

## Schema Enhancement

* We used the same schema used in Phase 1

## Memory and Cache Management Enhancement

* We used stored procedures to enhance memory and cache management

## Indexes Modifications

* We added some non-clustered inexes to speed up the data selection as follows:
  + Non-clustered index on: Ssn (Employee Table)
  + Non-clustered index on: Pnumber (Project Table)
  + Non-clustered index on: Bdate (Works\_on Table)
  + Non-clustered index on: Salary (Employee Table)

## Query Rewriting Modifications

* We rewrote the queries to replace the nested loops with INNER JOIN to make the query execute faster

# Conclusion

* Using SQL server is recommended for relational schemas as NoSQL has worse performance
* Index tuning has a great effect in the execution time for executing the queries
* Optimization for SQL query has a more noticeable effect for larger database sizes
* Stored procedures did not yield the expected optimization
* Seeing the execution plan and the index scan is very useful when tryin to generate non-clustered indexes to convert the scanning operations into seeking operations.