Credit Hours System

CMPN401 ADB

Cairo University

Faculty of Engineering

Advanced Database Project (Phase-2)

Group (14) Section (2)

Group Members:

1- Ahmed Elgarf 1180262

2- Mark Adel 1180495

3- Abdelrahman Amr 1180156

Submitted to:

1- Eng Marwa

2- Eng Ahmed Amrawy

Contents

System Information: ... 3

Query 1 ... 4

Before Optimization: ... 4

After Optimization: ... 5

Optimization Done: ... 5

Theoretical Parallel Query Processing Report: ... 6

Query 2 ... 6

Before Optimization: ... 6

After Optimization: ... 7

Optimization Done: ... 7

Theoretical Parallel Query Processing Report: ... 8

Query 3 ... 8

Before Optimization: ... 8

Optimization Done: ... 9

Theoretical Parallel Query Processing Report: ... 9

New Database Statistics: ... 10

Time Analysis: ... 11

Query1: ... 11

Query2: ... 12

Query3: ... 12

Times for different database sizes: ... 13

NOSQL Queries: ... 14

Query1: ... 14

Query2: ... 15

Query3: ... 16

Optimizations: ... 17

The Schema enhancement: ... 17

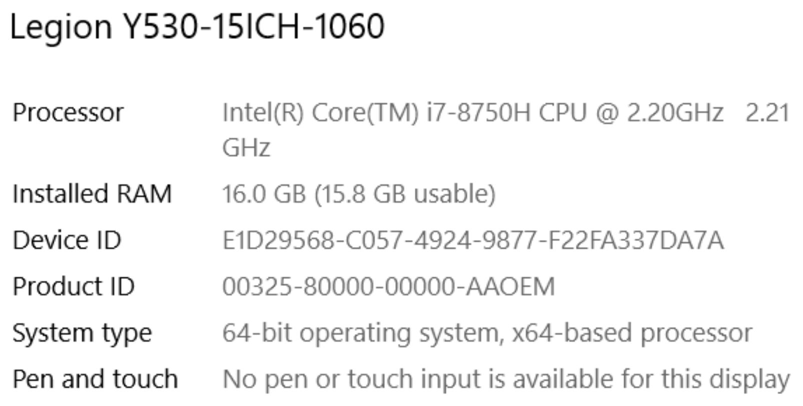
The memory management enhancement: ... 17

The Indexes modifications: ... 17

The Query rewriting Modifications: ... 17

Conclusion: ... 18

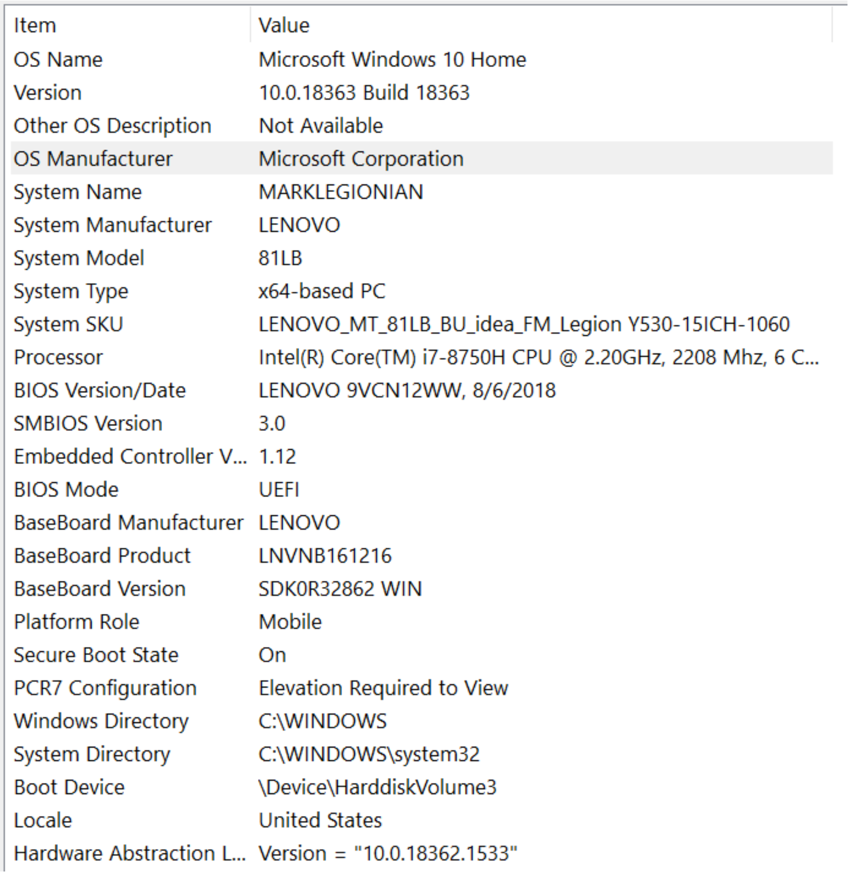
System Information:



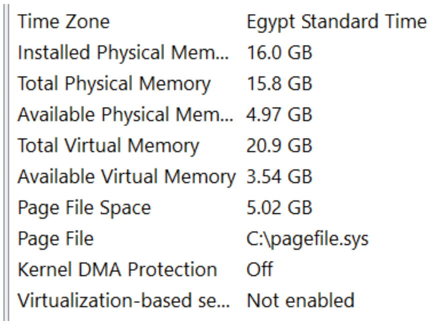
Legion Y530-15ICH-1060

Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz
Installed RAM	16.0 GB (15.8 GB usable)
Device ID	E1D29568-C057-4924-9877-F22FA337DA7A
Product ID	00325-80000-00000-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Hard Disk 2 TB HDD + 256 GB SSD



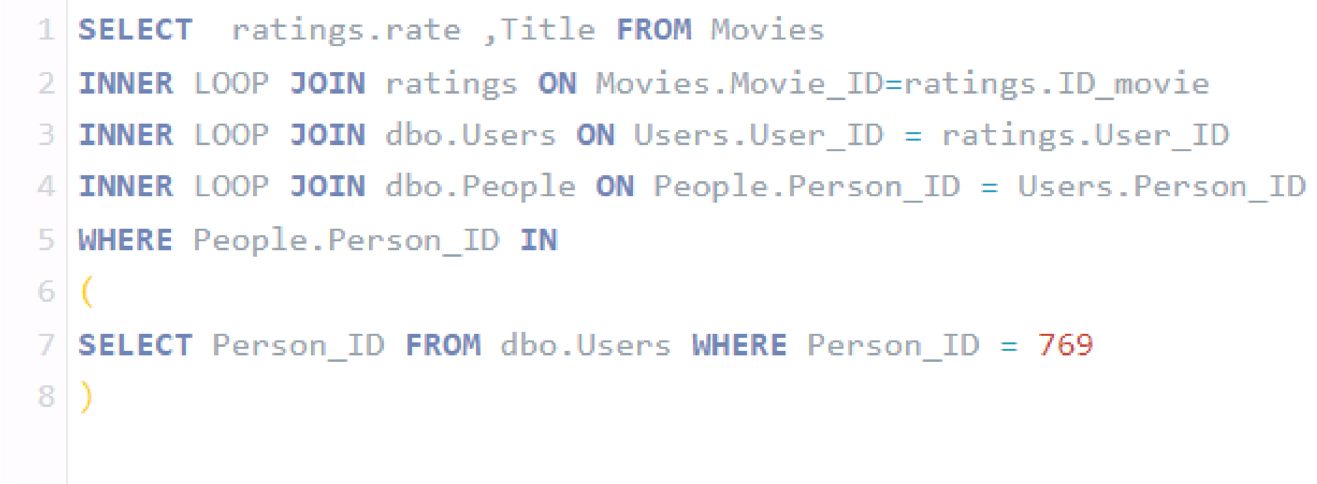
Item	Value
OS Name	Microsoft Windows 10 Home
Version	10.0.18363 Build 18363
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	MARKLEGIONIAN
System Manufacturer	LENOVO
System Model	81LB
System Type	x64-based PC
System SKU	LENOVO_MT_81LB_BU_idea_FM_Legion Y530-15ICH-1060
Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 2208 Mhz, 6 C...
BIOS Version/Date	LENOVO 9VCN12WW, 8/6/2018
SMBIOS Version	3.0
Embedded Controller V...	1.12
BIOS Mode	UEFI
BaseBoard Manufacturer	LENOVO
BaseBoard Product	LNVNB161216
BaseBoard Version	SDK0R32862 WIN
Platform Role	Mobile
Secure Boot State	On
PCR7 Configuration	Elevation Required to View
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume3
Locale	United States
Hardware Abstraction L...	Version = "10.0.18362.1533"



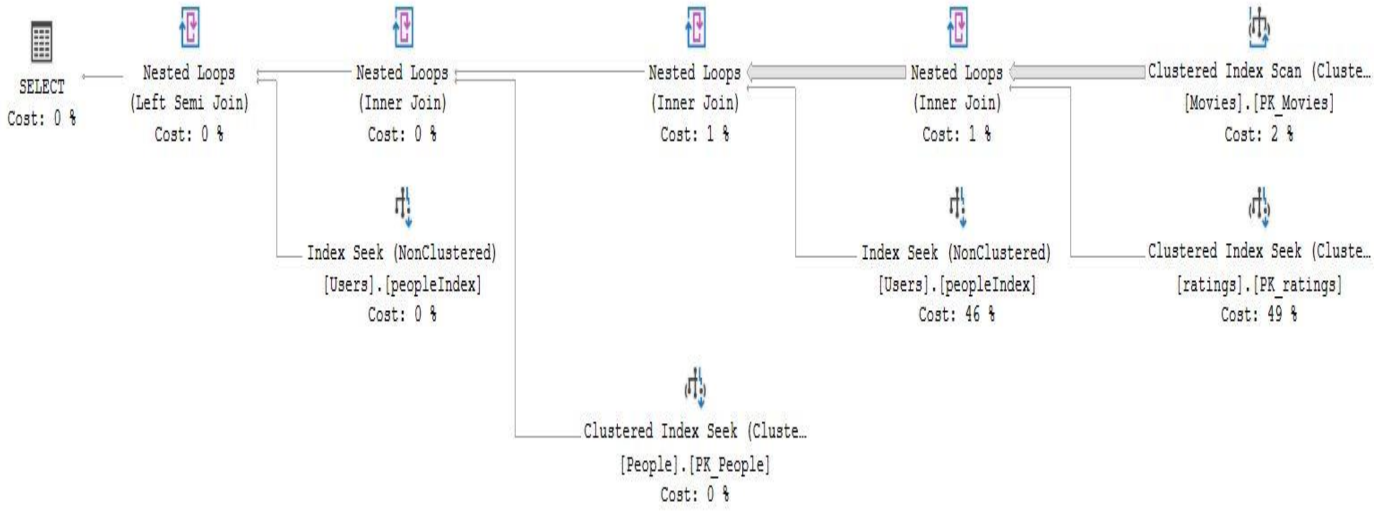
Time Zone	Egypt Standard Time
Installed Physical Mem...	16.0 GB
Total Physical Memory	15.8 GB
Available Physical Mem...	4.97 GB
Total Virtual Memory	20.9 GB
Available Virtual Memory	3.54 GB
Page File Space	5.02 GB
Page File	C:\pagefile.sys
Kernel DMA Protection	Off
Virtualization-based se...	Not enabled

Query 1

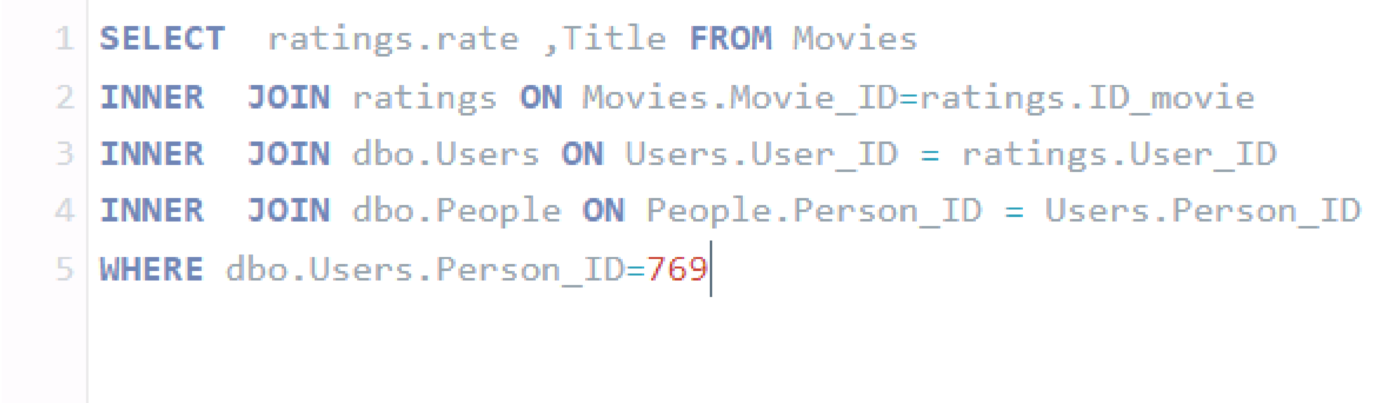
Before Optimization:



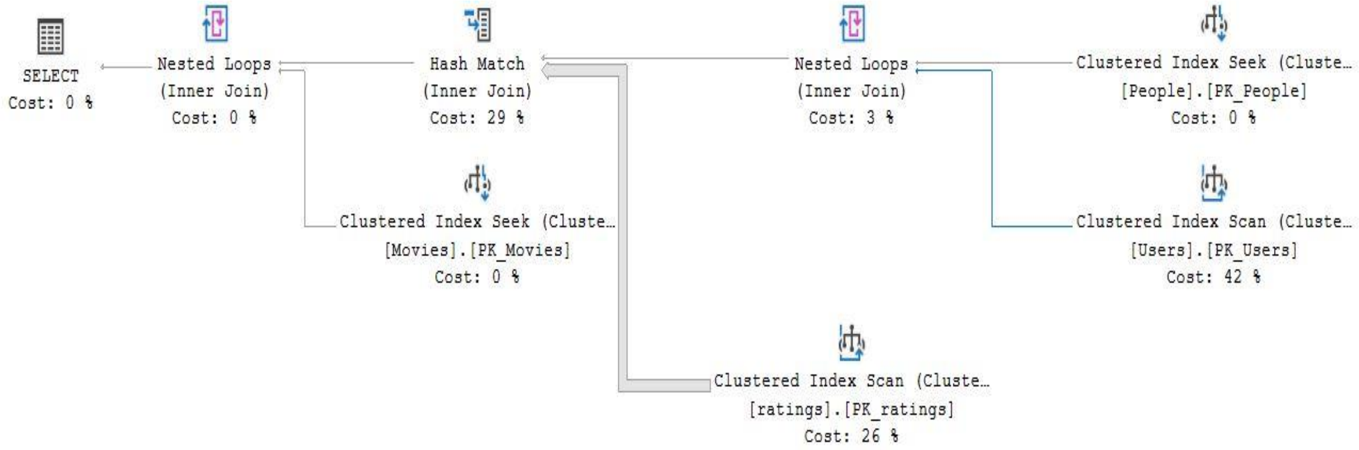
```
SELECT  ratings.rate ,Title FROM Movies
INNER LOOP JOIN ratings ON Movies.Movie_ID=ratings.ID_movie
INNER LOOP JOIN dbo.Users ON Users.User_ID = ratings.User_ID
INNER LOOP JOIN dbo.People ON People.Person_ID = Users.Person_ID
WHERE People.Person_ID IN
(
SELECT Person_ID FROM dbo.Users WHERE Person_ID = 769
)
```



After Optimization:



```sql
SELECT  ratings.rate ,Title FROM Movies
INNER  JOIN ratings ON Movies.Movie_ID=ratings.ID_movie
INNER  JOIN dbo.Users ON Users.User_ID = ratings.User_ID
INNER  JOIN dbo.People ON People.Person_ID = Users.Person_ID
WHERE dbo.Users.Person_ID=769
```



Optimization Done:

- Inner Join Instead of Nested loop join
- Non Clustered Index on : Person_ID
- Non Clustered Index on : User_type

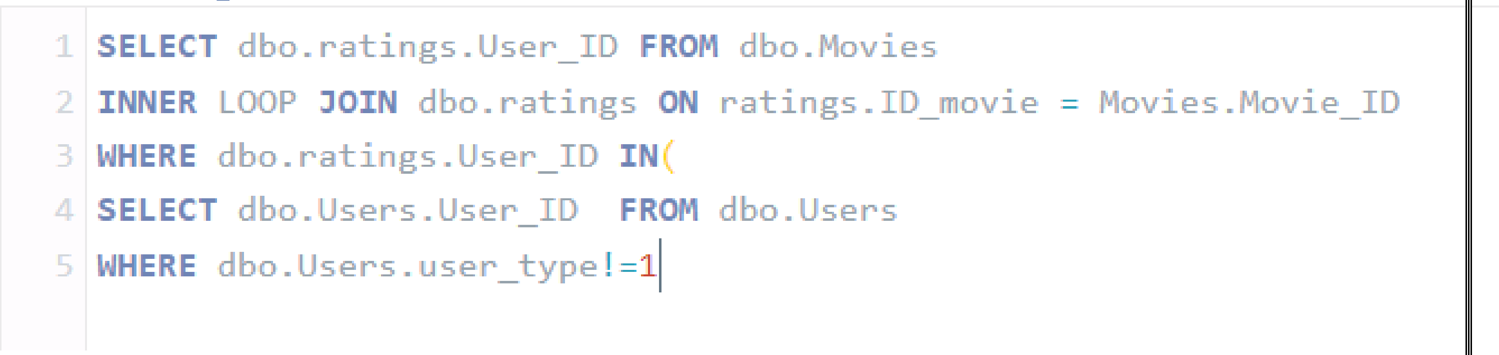
Action	CPU time	Elapsed time
Query 1 (no optimization)	3078 ms	3098
Query 1 (optimized inner joins)	156 ms	160ms
Non- cluster added with Optimization	94 ms	106 ms
Mongo	8321 ms	

Theoretical Parallel Query Processing Report:

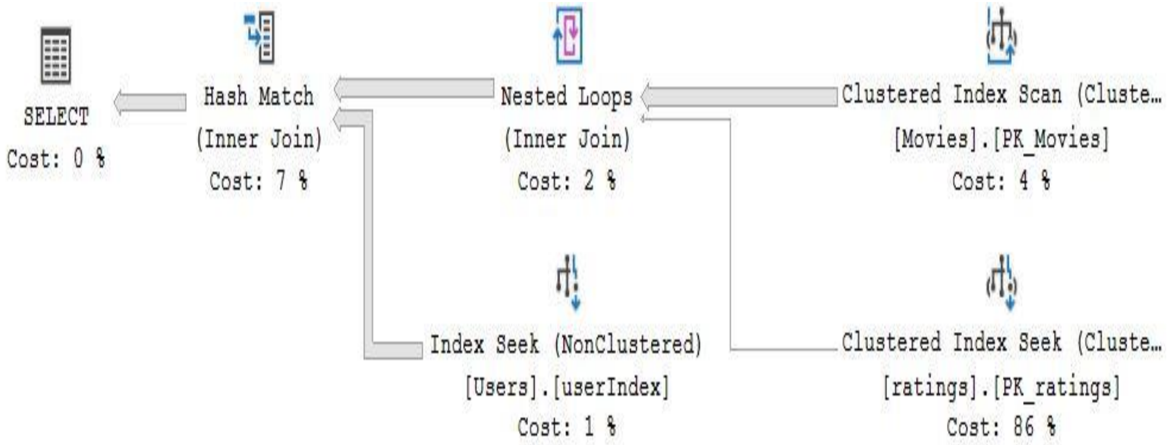
- Clustered index scan on Movies table
- Clustered index scan on Users table
- Clustered index scan on People.

Query 2

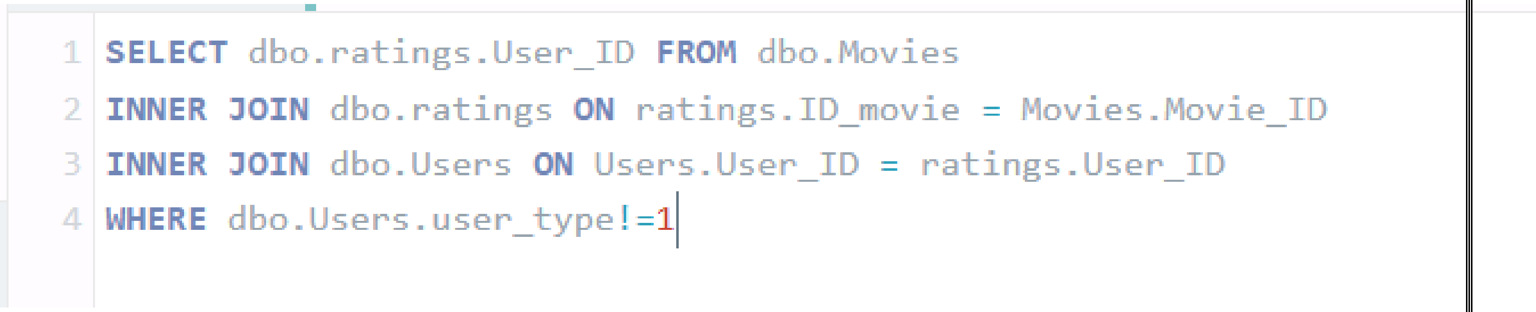
Before Optimization:



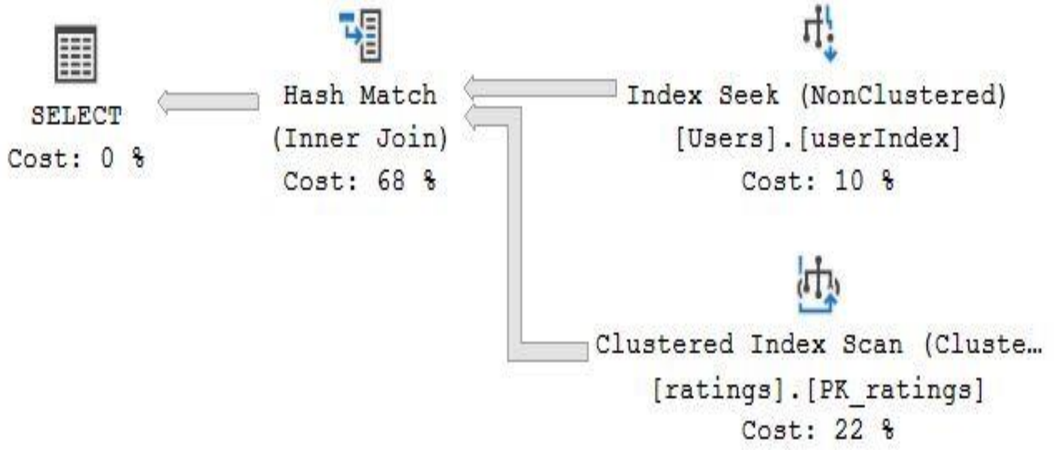
```
SELECT dbo.ratings.User_ID FROM dbo.Movies
INNER LOOP JOIN dbo.ratings ON ratings.ID_movie = Movies.Movie_ID
WHERE dbo.ratings.User_ID IN(
SELECT dbo.Users.User_ID  FROM dbo.Users
WHERE dbo.Users.user_type!=1
```



After Optimization:



```sql
SELECT dbo.ratings.User_ID FROM dbo.Movies
INNER JOIN dbo.ratings ON ratings.ID_movie = Movies.Movie_ID
INNER JOIN dbo.Users ON Users.User_ID = ratings.User_ID
WHERE dbo.Users.user_type!=1
```



Optimization Done:

- Inner Join Instead of Nested loop join
- Non-Clustered Index on: User_type

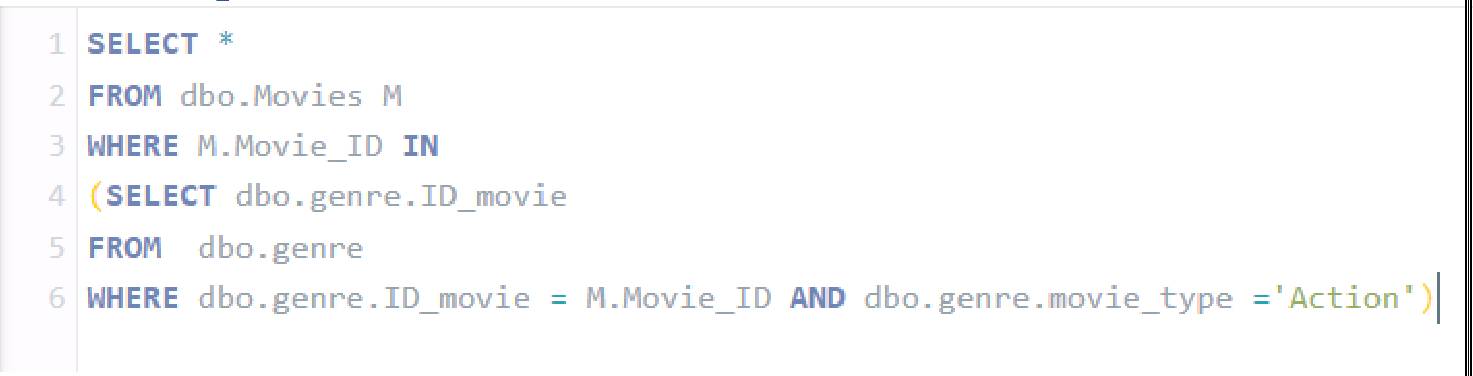
Action	CPU time	Elapsed time
Query 1 (no optimization)	2328 ms	7416 ms
Query 1 (optimized inner joins)	687 ms	5587 ms
Non- cluster index added with Optimization	563 ms	5612 ms
Mongo	30,906	

Theoretical Parallel Query Processing Report:

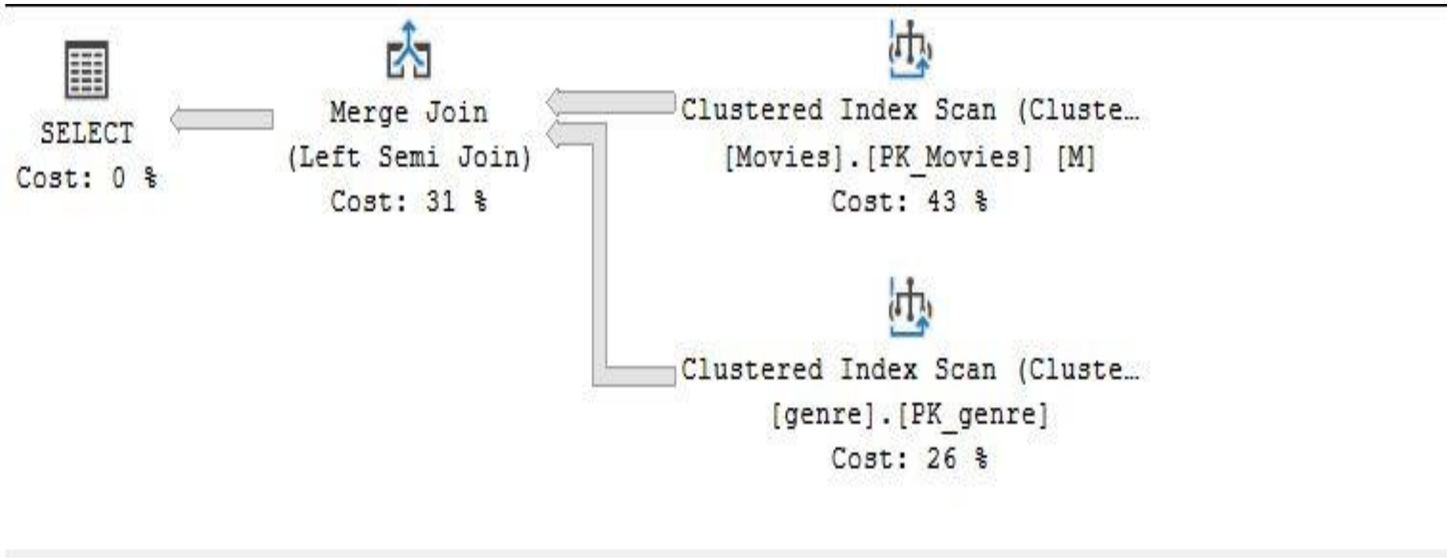
- Clustered index scan on Movies table
- Clustered index scan on Users table
- Clustered index scan on Users.

Query 3

Before Optimization:



```
SELECT *
FROM dbo.Movies M
WHERE M.Movie_ID IN
(SELECT dbo.genre.ID_movie
FROM  dbo.genre
WHERE dbo.genre.ID_movie = M.Movie_ID AND dbo.genre.movie_type ='Action')
```



Optimization Done:

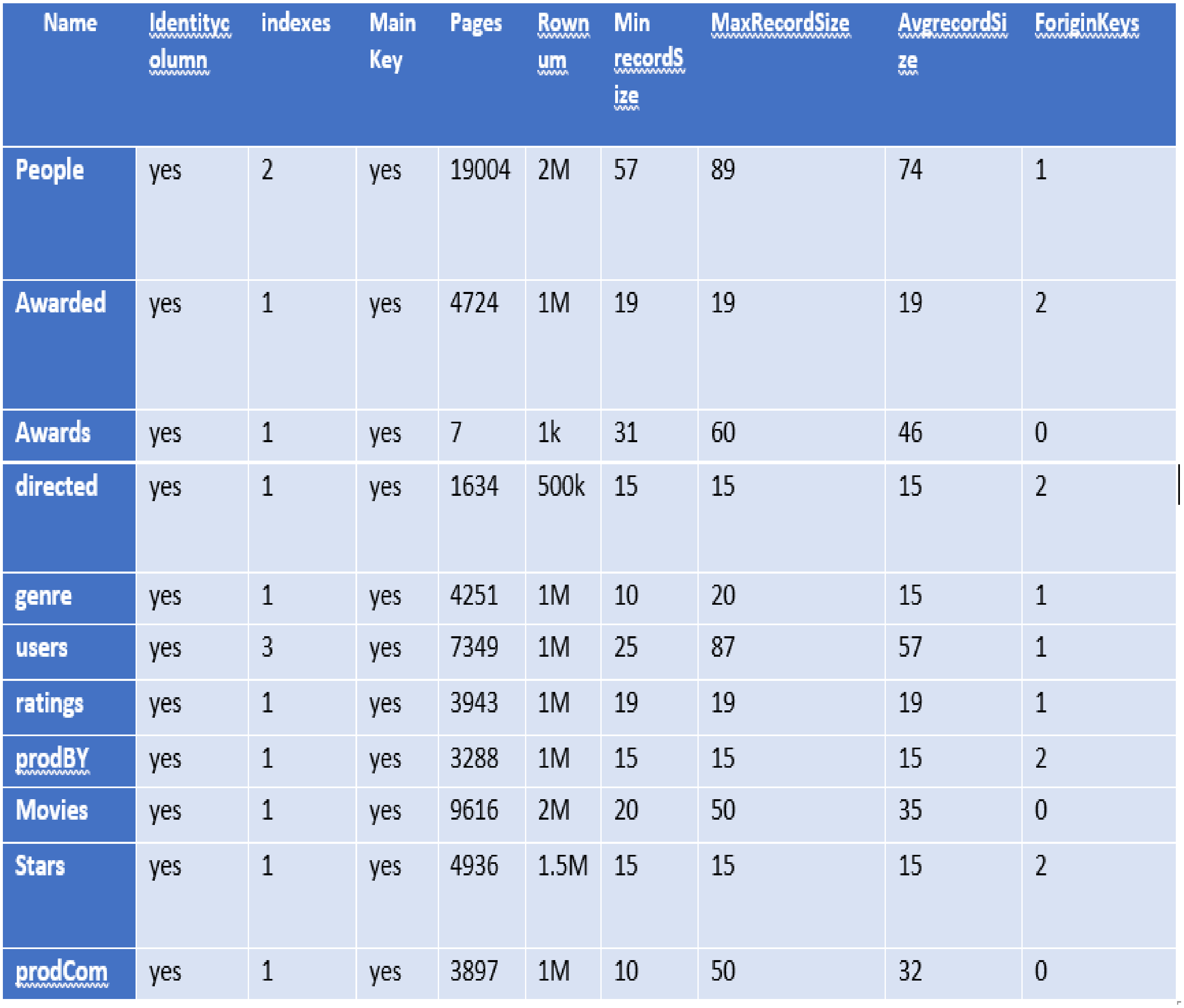
- Changed title 50 --> 20 , lang 50 --->20 , composer 100 ----> 20, country (50 ----> 20) genre movie type (50 -----> 20)

Action	CPU time	Elapsed time
Query 1 (no optimization)	1657 ms	10886ms
Query 1 (optimized changed varchar)	1141 ms	10334 ms
Mongo	143,875 ms	

Theoretical Parallel Query Processing Report:

- Clustered index scan on Movies table
- Clustered index scan on Genre table

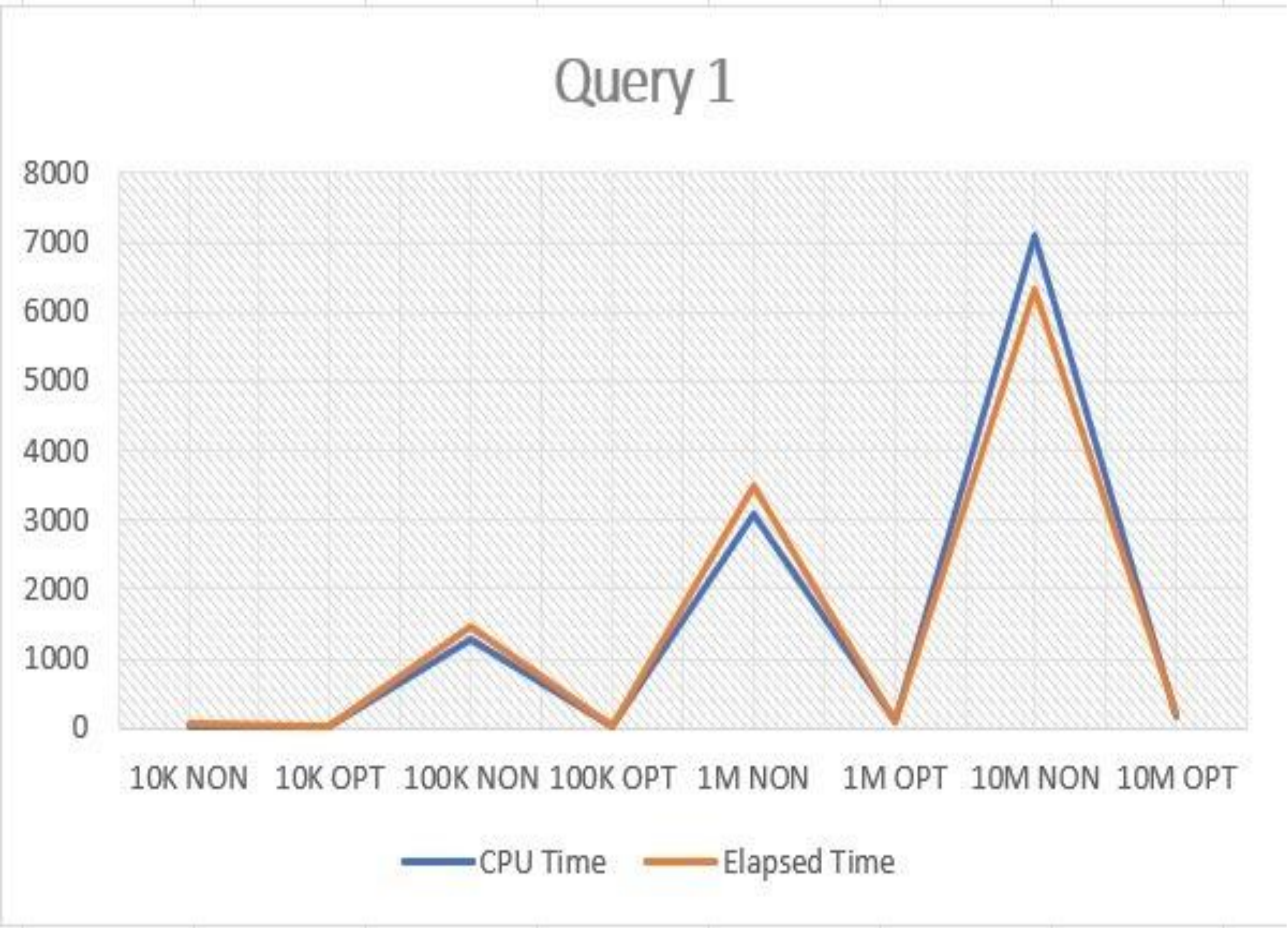
New Database Statistics:



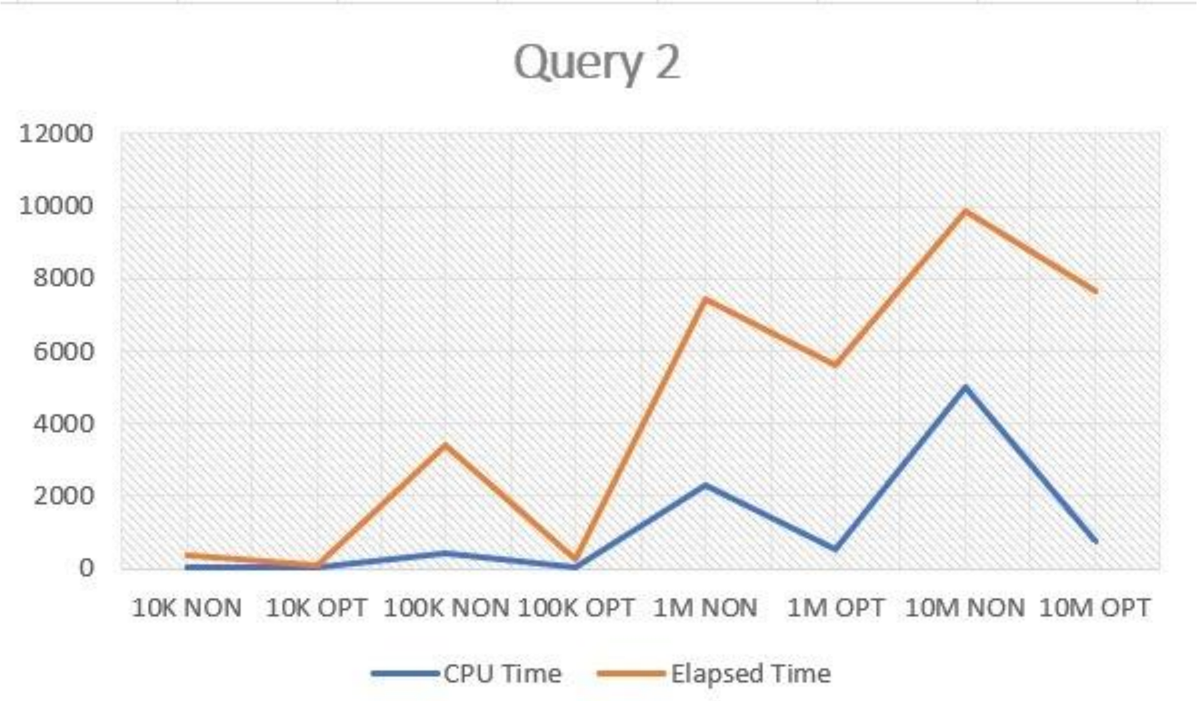
Name	Identitycolumn	indexes	Main Key	Pages	Rownum	Min recordSize	MaxRecordSize	AvgrecordSize	ForiginKeys
People	yes	2	yes	19004	2M	57	89	74	1
Awarded	yes	1	yes	4724	1M	19	19	19	2
Awards	yes	1	yes	7	1k	31	60	46	0
directed	yes	1	yes	1634	500k	15	15	15	2
genre	yes	1	yes	4251	1M	10	20	15	1
users	yes	3	yes	7349	1M	25	87	57	1
ratings	yes	1	yes	3943	1M	19	19	19	1
prodBY	yes	1	yes	3288	1M	15	15	15	2
Movies	yes	1	yes	9616	2M	20	50	35	0
Stars	yes	1	yes	4936	1.5M	15	15	15	2
prodCom	yes	1	yes	3897	1M	10	50	32	0

Time Analysis:

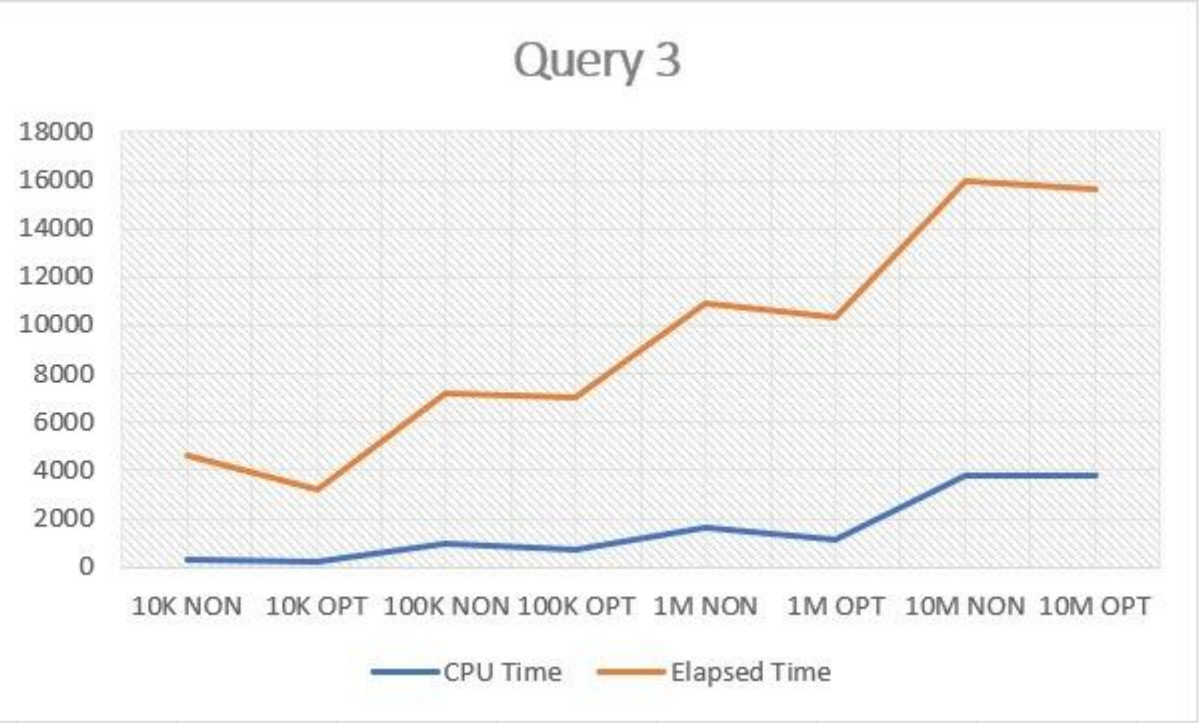
Query1:



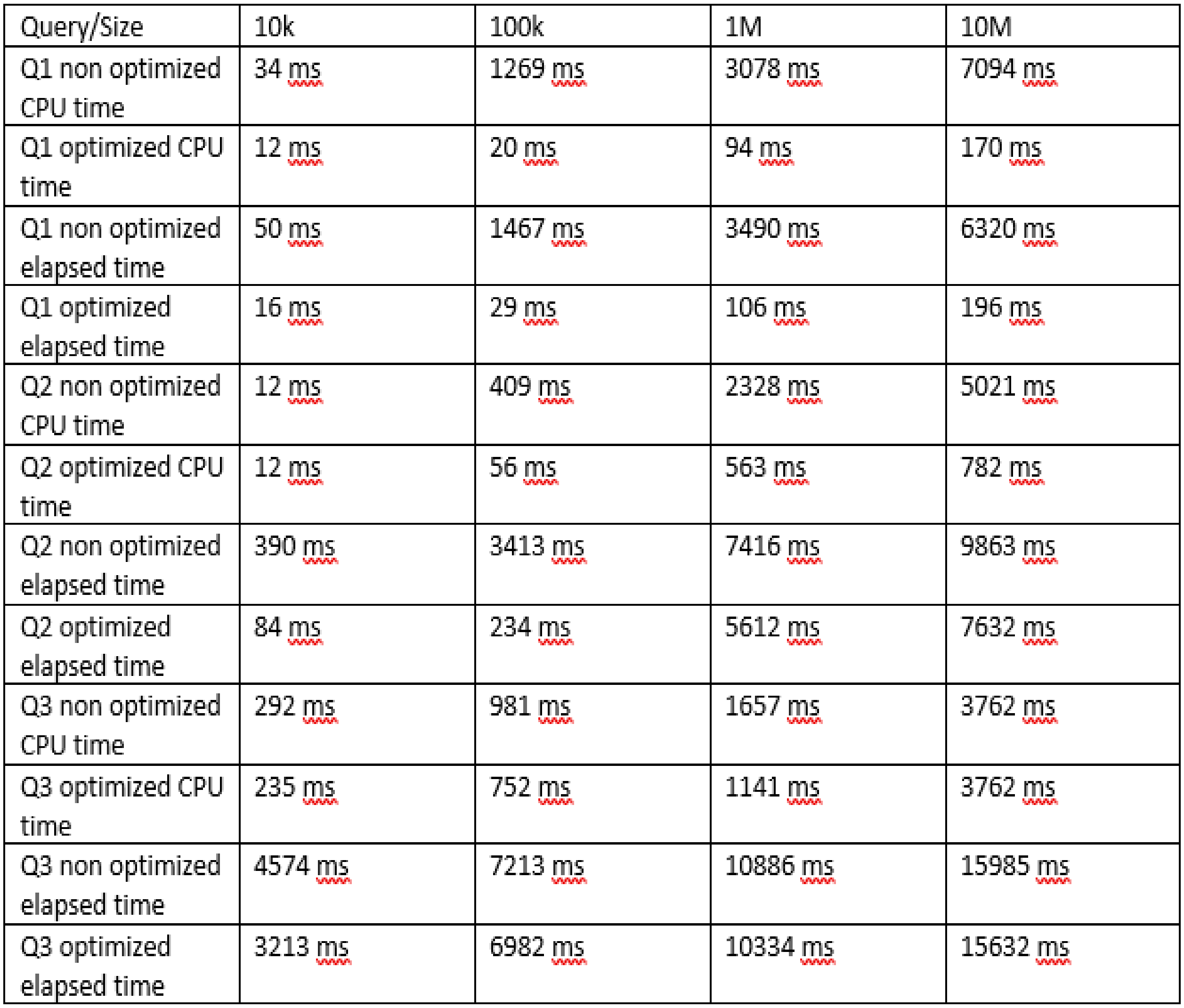
Query2:



Query3:



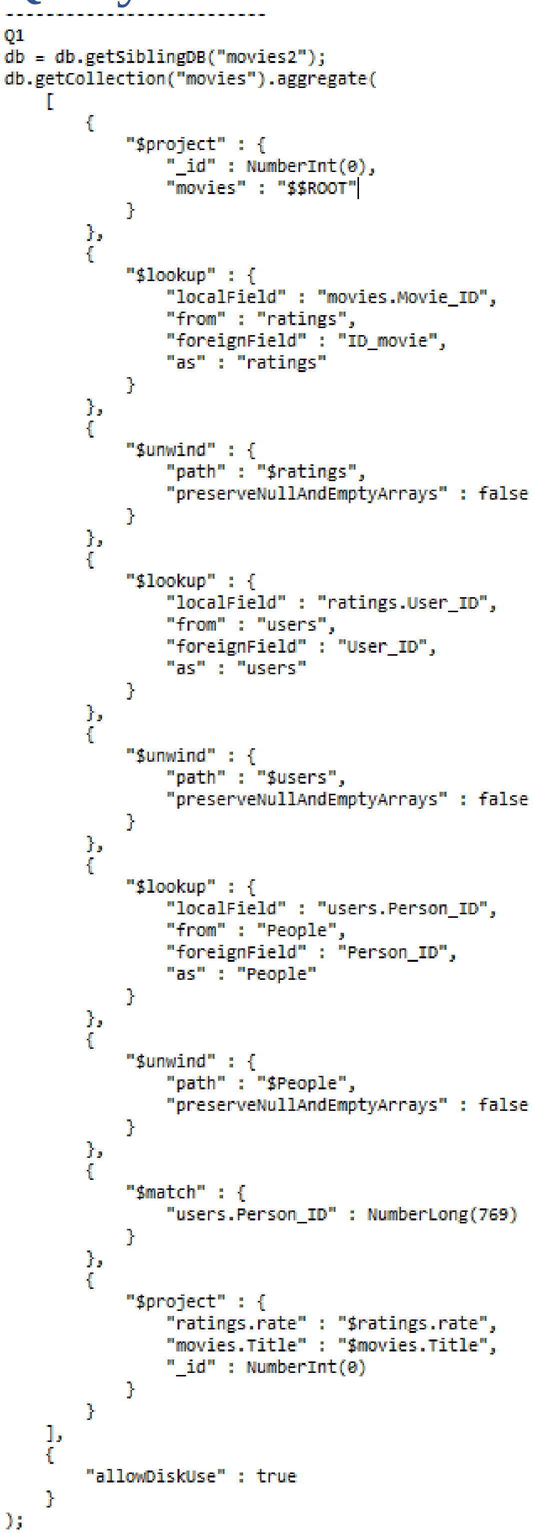
Times for different database sizes:



Query/Size	10k	100k	1M	10M
Q1 non optimized CPU time	34 ms	1269 ms	3078 ms	7094 ms
Q1 optimized CPU time	12 ms	20 ms	94 ms	170 ms
Q1 non optimized elapsed time	50 ms	1467 ms	3490 ms	6320 ms
Q1 optimized elapsed time	16 ms	29 ms	106 ms	196 ms
Q2 non optimized CPU time	12 ms	409 ms	2328 ms	5021 ms
Q2 optimized CPU time	12 ms	56 ms	563 ms	782 ms
Q2 non optimized elapsed time	390 ms	3413 ms	7416 ms	9863 ms
Q2 optimized elapsed time	84 ms	234 ms	5612 ms	7632 ms
Q3 non optimized CPU time	292 ms	981 ms	1657 ms	3762 ms
Q3 optimized CPU time	235 ms	752 ms	1141 ms	3762 ms
Q3 non optimized elapsed time	4574 ms	7213 ms	10886 ms	15985 ms
Q3 optimized elapsed time	3213 ms	6982 ms	10334 ms	15632 ms

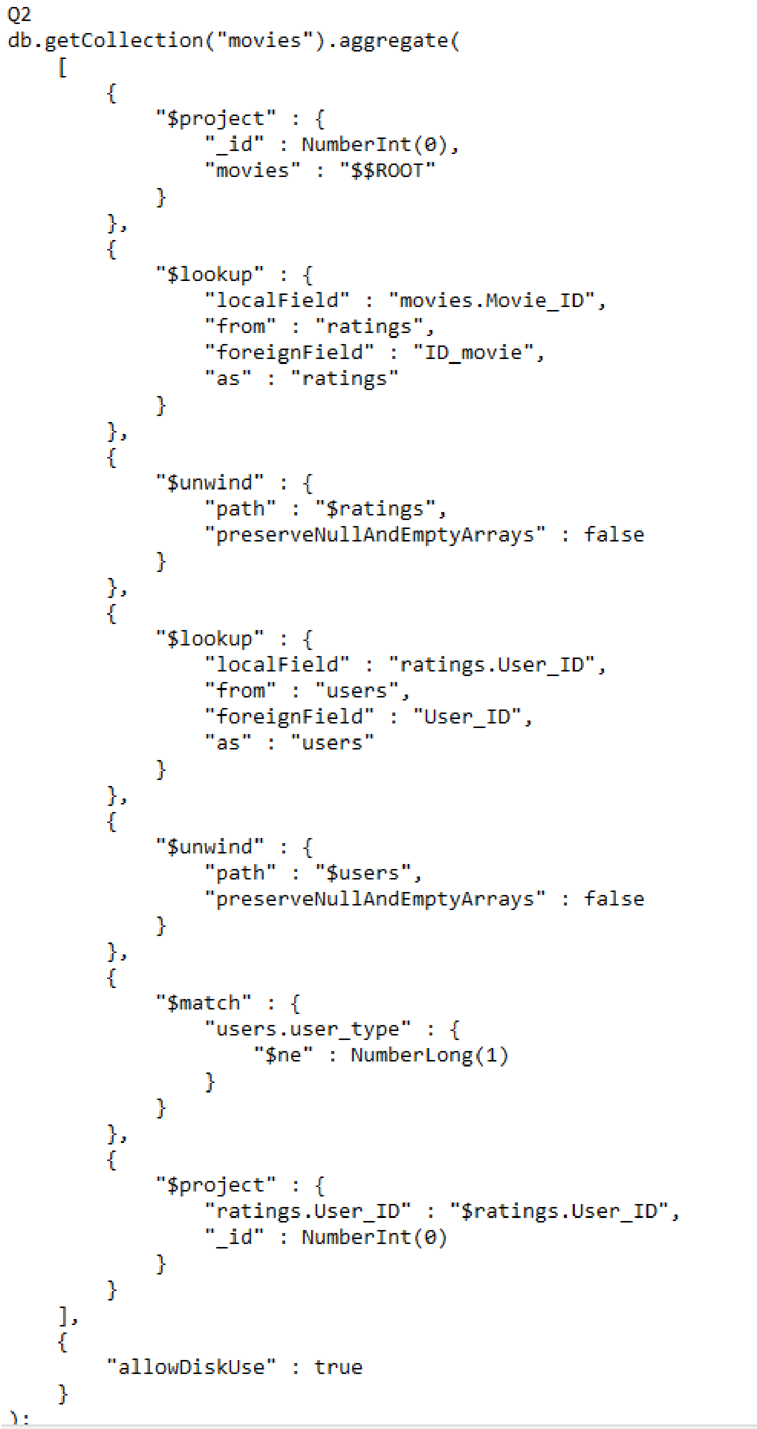
NOSQL Queries:

Query1:



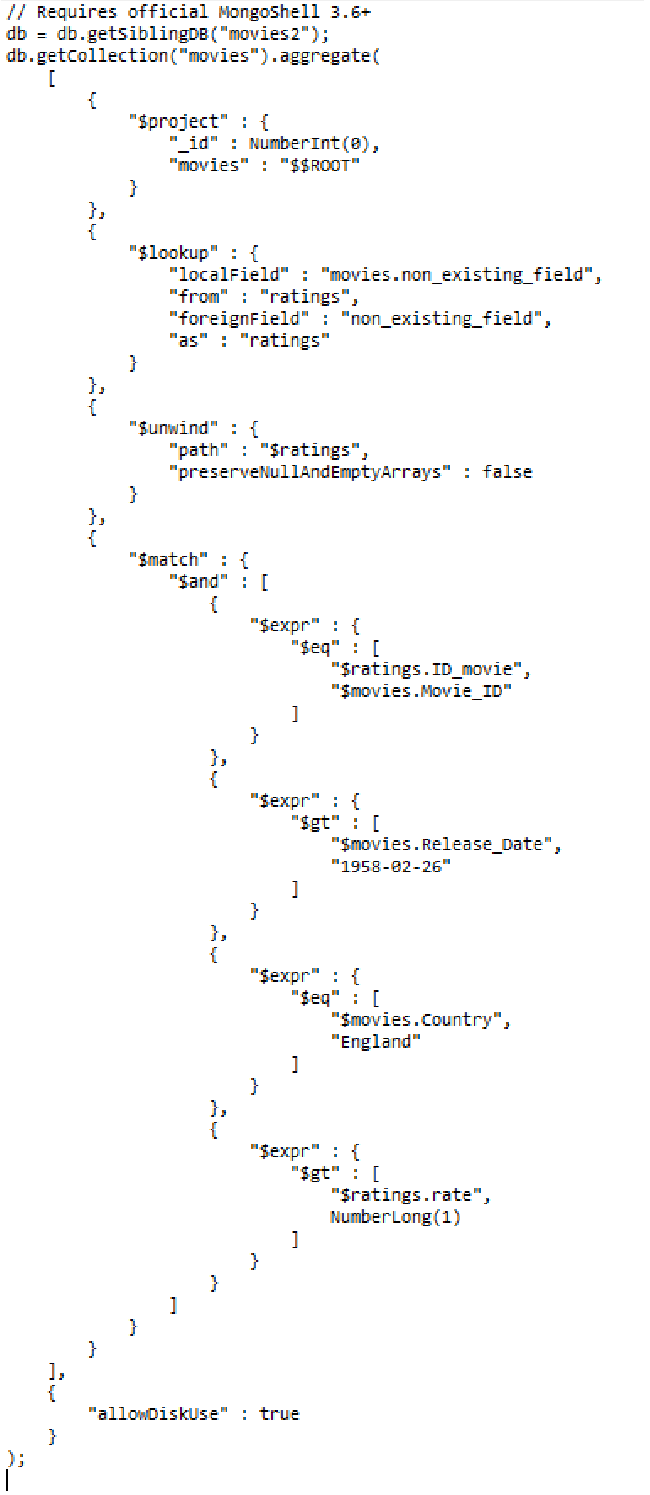
```
--------------------------
Q1
db = db.getSiblingDB("movies2");
db.getCollection("movies").aggregate(
    [
        {
            "$project" : {
                "_id" : NumberInt(0),
                "movies" : "$$ROOT"
            }
        },
        {
            "$lookup" : {
                "localField" : "movies.Movie_ID",
                "from" : "ratings",
                "foreignField" : "ID_movie",
                "as" : "ratings"
            }
        },
        {
            "$unwind" : {
                "path" : "$ratings",
                "preserveNullAndEmptyArrays" : false
            }
        },
        {
            "$lookup" : {
                "localField" : "ratings.User_ID",
                "from" : "users",
                "foreignField" : "User_ID",
                "as" : "users"
            }
        },
        {
            "$unwind" : {
                "path" : "$users",
                "preserveNullAndEmptyArrays" : false
            }
        },
        {
            "$lookup" : {
                "localField" : "users.Person_ID",
                "from" : "People",
                "foreignField" : "Person_ID",
                "as" : "People"
            }
        },
        {
            "$unwind" : {
                "path" : "$People",
                "preserveNullAndEmptyArrays" : false
            }
        },
        {
            "$match" : {
                "users.Person_ID" : NumberLong(769)
            }
        },
        {
            "$project" : {
                "ratings.rate" : "$ratings.rate",
                "movies.Title" : "$movies.Title",
                "_id" : NumberInt(0)
            }
        }
    ],
    {
        "allowDiskUse" : true
    }
);
```

Query2:



```
Q2
db.getCollection("movies").aggregate(
    [
        {
            "$project" : {
                "_id" : NumberInt(0),
                "movies" : "$$ROOT"
            }
        },
        {
            "$lookup" : {
                "localField" : "movies.Movie_ID",
                "from" : "ratings",
                "foreignField" : "ID_movie",
                "as" : "ratings"
            }
        },
        {
            "$unwind" : {
                "path" : "$ratings",
                "preserveNullAndEmptyArrays" : false
            }
        },
        {
            "$lookup" : {
                "localField" : "ratings.User_ID",
                "from" : "users",
                "foreignField" : "User_ID",
                "as" : "users"
            }
        },
        {
            "$unwind" : {
                "path" : "$users",
                "preserveNullAndEmptyArrays" : false
            }
        },
        {
            "$match" : {
                "users.user_type" : {
                    "$ne" : NumberLong(1)
                }
            }
        },
        {
            "$project" : {
                "ratings.User_ID" : "$ratings.User_ID",
                "_id" : NumberInt(0)
            }
        }
    ],
    {
        "allowDiskUse" : true
    }
);
```

Query3:



```
// Requires official MongoShell 3.6+
db = db.getSiblingDB("movies2");
db.getCollection("movies").aggregate(
    [
        {
            "$project" : {
                "_id" : NumberInt(0),
                "movies" : "$$ROOT"
            }
        },
        {
            "$lookup" : {
                "localField" : "movies.non_existing_field",
                "from" : "ratings",
                "foreignField" : "non_existing_field",
                "as" : "ratings"
            }
        },
        {
            "$unwind" : {
                "path" : "$ratings",
                "preserveNullAndEmptyArrays" : false
            }
        },
        {
            "$match" : {
                "$and" : [
                    {
                        "$expr" : {
                            "$eq" : [
                                "$ratings.ID_movie",
                                "$movies.Movie_ID"
                            ]
                        }
                    },
                    {
                        "$expr" : {
                            "$gt" : [
                                "$movies.Release_Date",
                                "1958-02-26"
                            ]
                        }
                    },
                    {
                        "$expr" : {
                            "$eq" : [
                                "$movies.Country",
                                "England"
                            ]
                        }
                    },
                    {
                        "$expr" : {
                            "$gt" : [
                                "$ratings.rate",
                                NumberLong(1)
                            ]
                        }
                    }
                ]
            }
        }
    ],
    {
        "allowDiskUse" : true
    }
);
```

Optimizations:

The Schema enhancement:

- No modifications are done in the schema.

The memory management enhancement:

- We reduced the size of some column as follow: we changed the following columns: Title, lang, composer, and country in table Movies from VARCHAR(50) to VARCHAR(20) and column genre movie type in Genre table from VARCHAR(50) to VARCHAR(20).

The Indexes modifications:

We added some non-clustered indexes to speed up the data selection as follow:

- Non Clustered Index on : Person_ID in people table
- Non Clustered Index on : User_type on users table

The Query rewriting Modifications:

- We rewrote the query and replace the inner loop join with inner join to make the query execute faster and take smaller time in execution.

Conclusion:

- We recommend using SQL server and execute SQL query as it has good optimizations that can be done to the queries itself like rewriting the query to execute faster.
- Index tuning has a great effect in the execution time for executing the queries.
- Optimization for SQL query has a noticeable effect when the size of the database is large.
- We recommend making memory optimization by changing the modify the type or changing the length as (VARCHAR(100) to VARCHAR(20)).
- It is very useful to see the execution plan and see the index scans and try to generate non clustered indexes to convert the scanning operations into seeking operations.
- NOSQL is not recommended as SQL has a better performance compared to it.