Faculty of engineering
Computer engineering dep.

Database Phase 2

Team Members

Name	Sec.	Bn.
Sarah Mohamed Ahmed lotfy	1	23
Salma Mohamed Ali	1	27
Kareem Omar	2	6

Queries:

Q1: return employee FName, employee LName and project name of employees that work on projects with id >20 then order by employee FName.

Q2: return project name and project description and department name that the manager of project's department has salary > 20 then order by project name.

Q3:return department name and department address of departments that its managers work on projects whose names != 'Hills' then order by department name.

We run SQL queries on microsoft sql server and run NOSQL queries on mongodb.

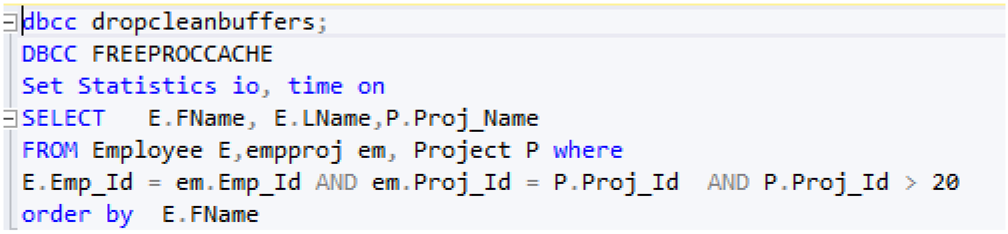
We used 5 optimization methods:

1. **Schema optimization**
2. **Index tuning**
3. **Memory and cache optimization: stored procedures**
4. **Query optimization using Joins**
5. **Partitioning**

1. Before and After Optimization Query Statistics

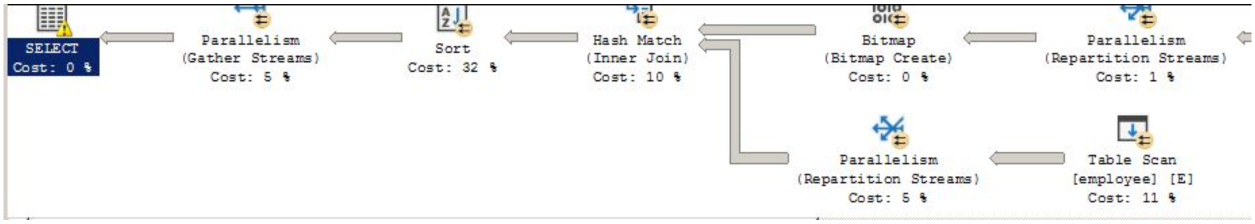
a) Execution plan for each query (Query tree)

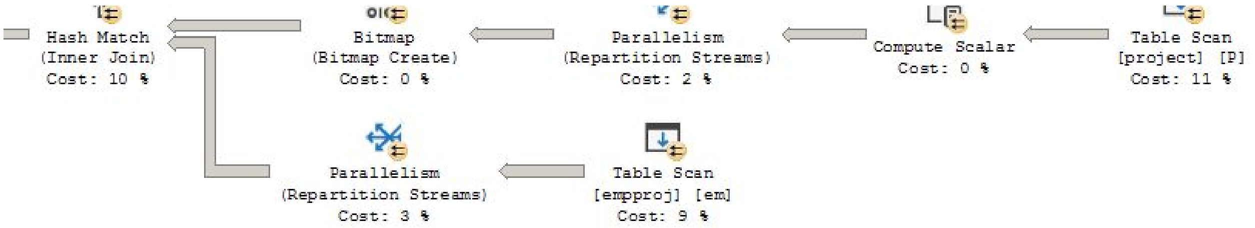
Query 1 return 999984 rows



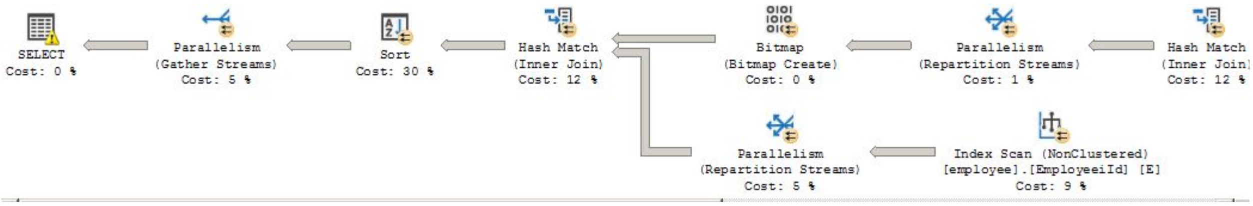
```
dbcc dropcleanbuffers;
DBCC FREEPROCCACHE
Set Statistics io, time on
SELECT   E.FName, E.LName,P.Proj_Name
FROM Employee E,empproj em, Project P where
E.Emp_Id = em.Emp_Id AND em.Proj_Id = P.Proj_Id  AND P.Proj_Id > 20
order by  E.FName
```

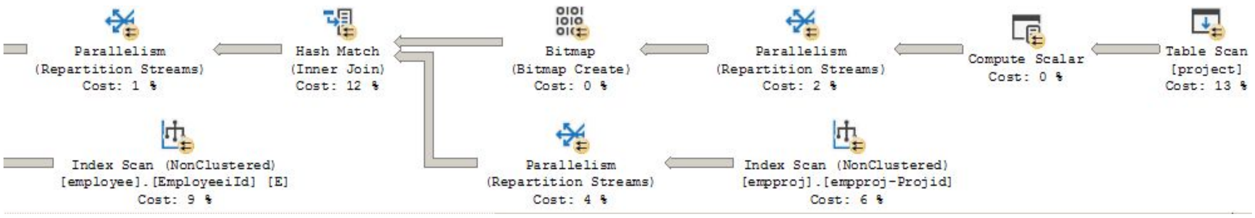
Query 1 before optimization



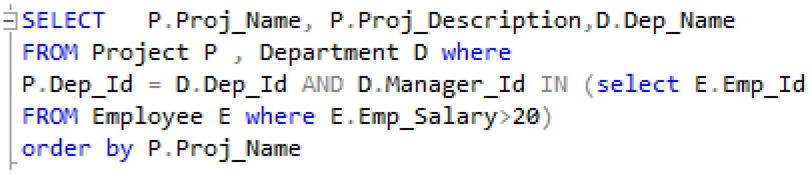


Query 1 after optimization



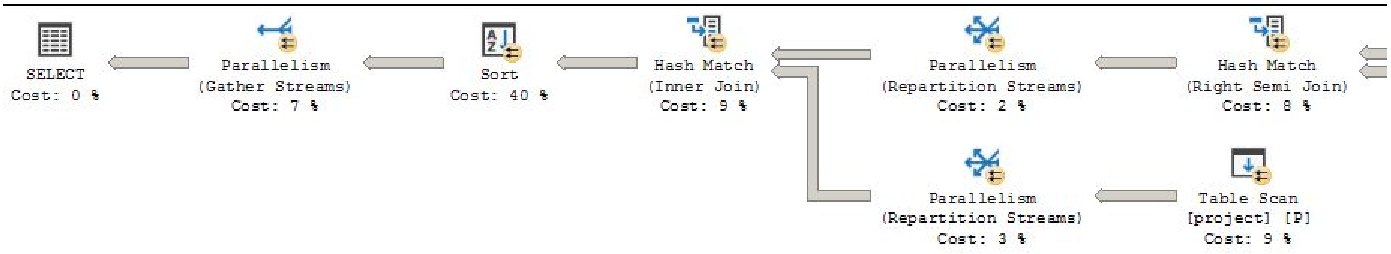


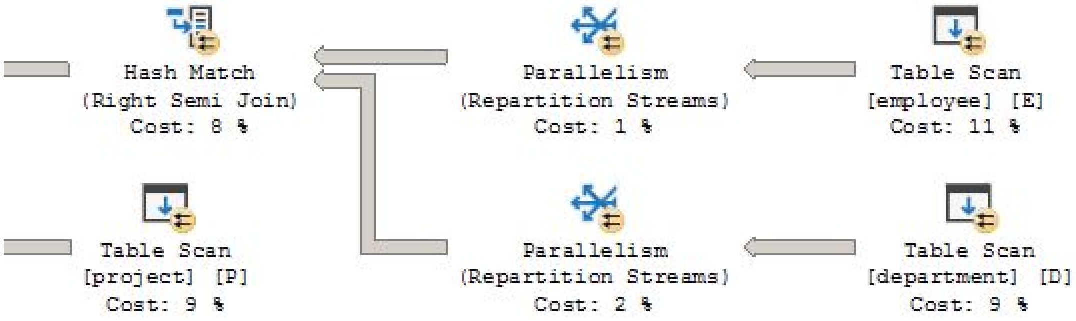
Query 2 : Return 1000000 rows



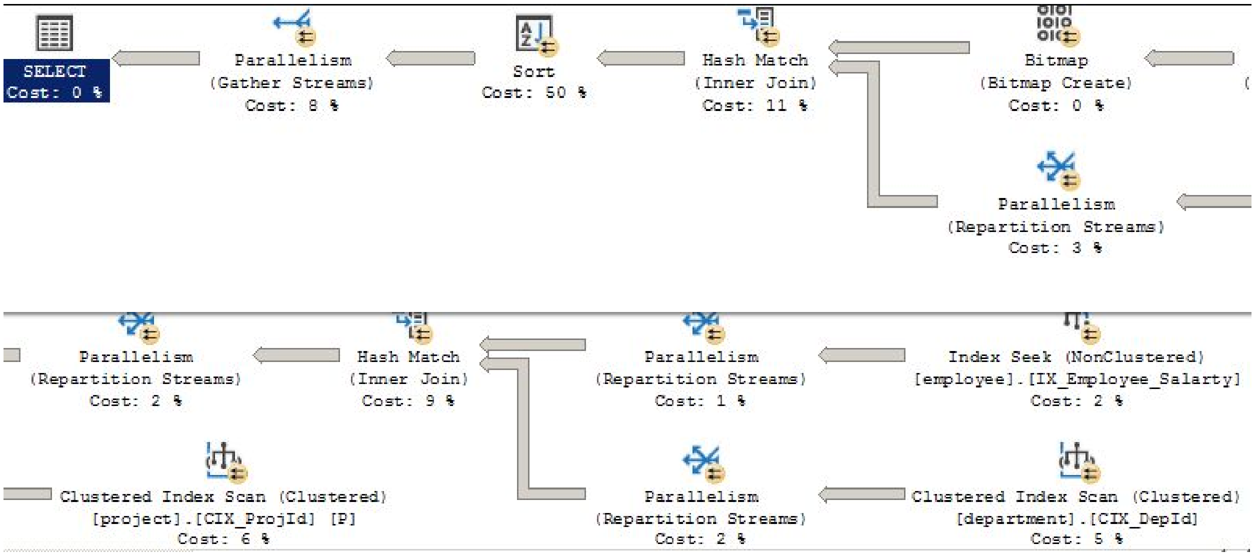
```
SELECT    P.Proj_Name, P.Proj_Description,D.Dep_Name
FROM Project P , Department D where
P.Dep_Id = D.Dep_Id AND D.Manager_Id IN (select E.Emp_Id
FROM Employee E where E.Emp_Salary>20)
order by P.Proj_Name
```

- **Query 2 Before optimization:**

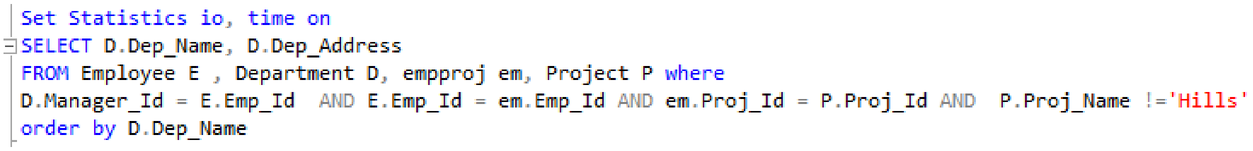




- **Query 2 After optimization:**

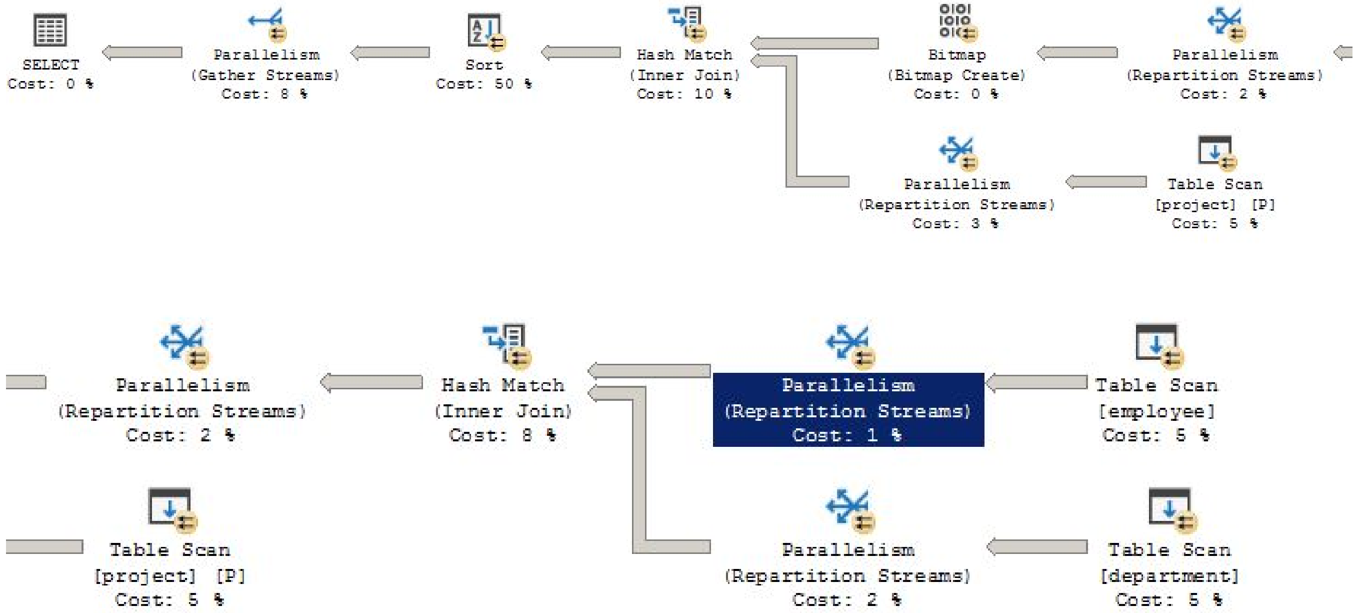


Query 3 return 999503 rows

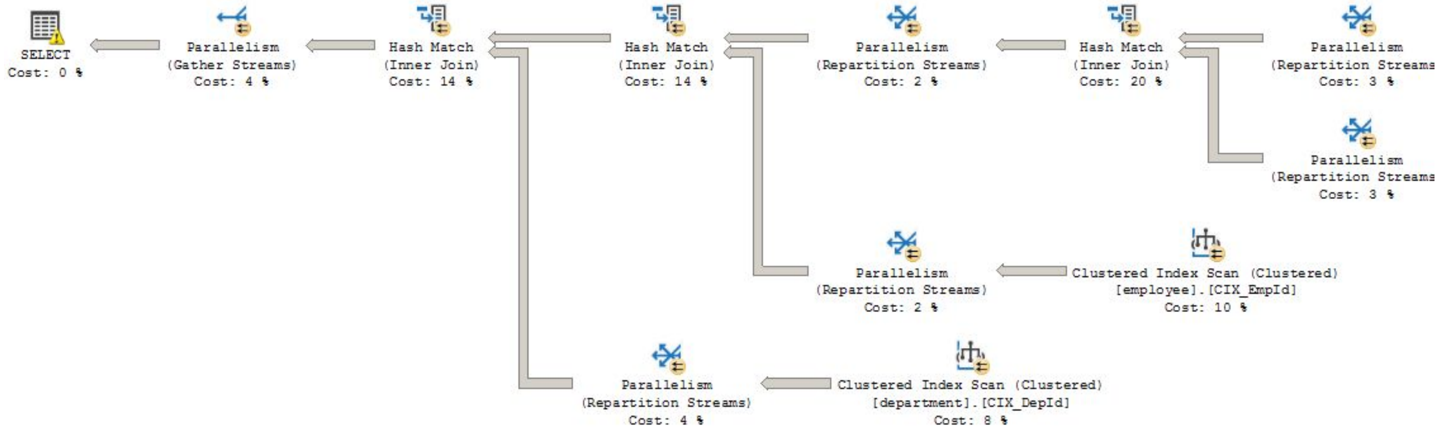


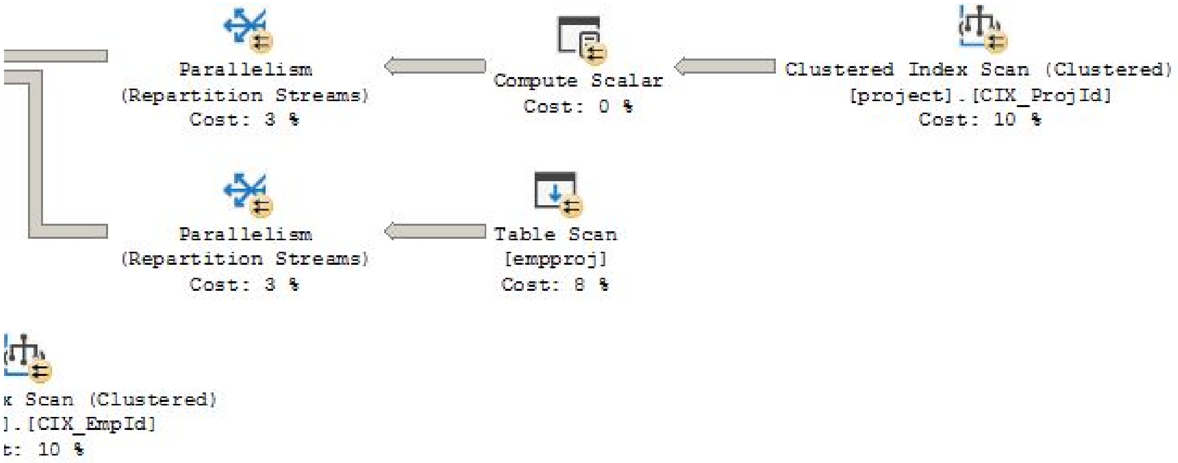
```
Set Statistics io, time on
SELECT D.Dep_Name, D.Dep_Address
FROM Employee E , Department D, empproj em, Project P where
D.Manager_Id = E.Emp_Id  AND E.Emp_Id = em.Emp_Id AND em.Proj_Id = P.Proj_Id AND  P.Proj_Name !='Hills'
order by D.Dep_Name
```

Query 3 Before optimization



Query 3 after optimization





2. Optimization Details

a) new database statistics after modification

Table name	Row count	Main key	Indexes	FK	Identity column	Max row size (Bytes)
Employee	1000000	yes	3	1	no	53
Department	1000000	yes	1	1	no	48
Project	1000000	yes	1	1	no	57
empproj	1000000	yes	2	2	no	8

b) The enhancement in the schema

- Changing the byte size of attributes of each table to take smaller size.
- A field should be specified explicitly as NOT NULL.
- Make Emp_Id the primary key for empproj table instead of the composite key.

Employee	Emp_Id (primary key)	Emp_FName	Emp_LName	Emp_Address	Emp_Salary	Dep_Id
	int	varchar(8)	varchar(8)	varchar(25)	int	int
	Not NULL	Not NULL	Not NULL	Not NULL	Not NULL	Not NULL

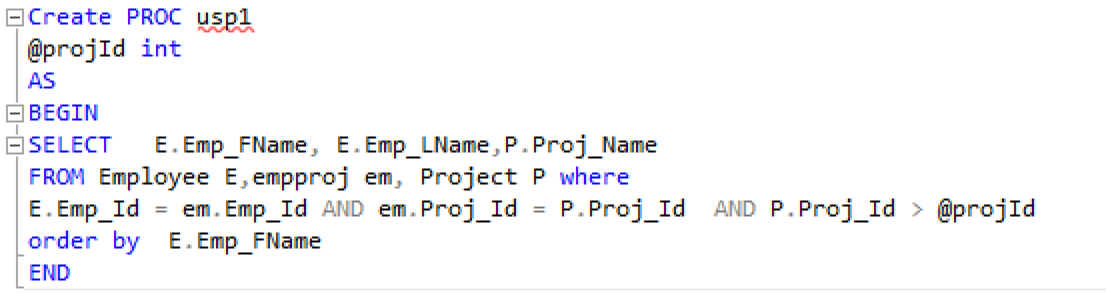
Department	Dep_Id (primary key)	Dep_Name	Dep_Address	Manager_Id
	int	varchar(15)	varchar(25)	int
	Not NULL	Not NULL	Not NULL	Not NULL

Project	Proj_Id (primary key)	Proj_Name	Proj_Description	Dep_Id
	int	varchar(20)	varchar(29)	int
	Not NULL	Not NULL	Not NULL	Not NULL

Empproj	Emp_Id (primary key)	Proj_Id
	int	int
	Not NULL	Not NULL

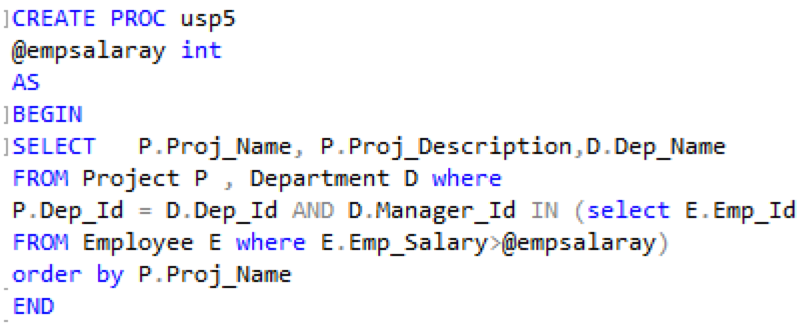
c) The enhancement in the memory management using stored procedures

Query 1

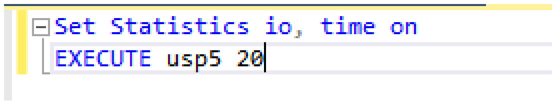


```
Create PROC usp1
@projId int
AS
BEGIN
SELECT   E.Emp_FName, E.Emp_LName,P.Proj_Name
FROM Employee E,empproj em, Project P where
E.Emp_Id = em.Emp_Id AND em.Proj_Id = P.Proj_Id  AND P.Proj_Id > @projId
order by  E.Emp_FName
END
```

Query 2

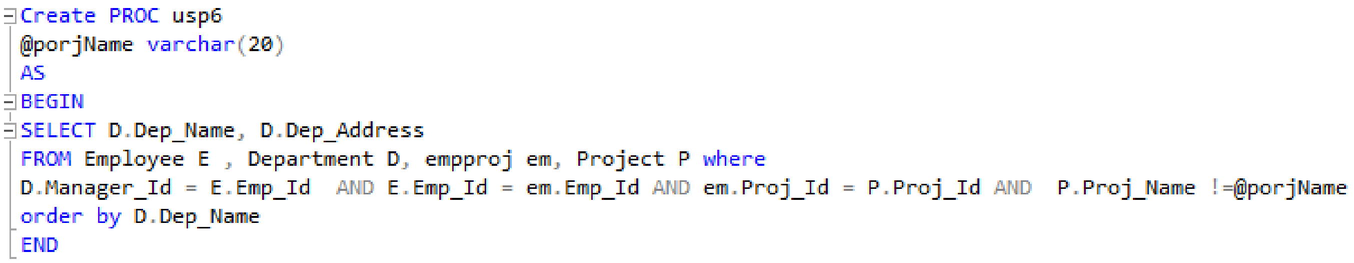


```
CREATE PROC usp5
@empsalaray int
AS
BEGIN
SELECT   P.Proj_Name, P.Proj_Description,D.Dep_Name
FROM Project P , Department D where
P.Dep_Id = D.Dep_Id AND D.Manager_Id IN (select E.Emp_Id
FROM Employee E where E.Emp_Salary>@empsalaray)
order by P.Proj_Name
END
```



```
Set Statistics io, time on
EXECUTE usp5 20
```

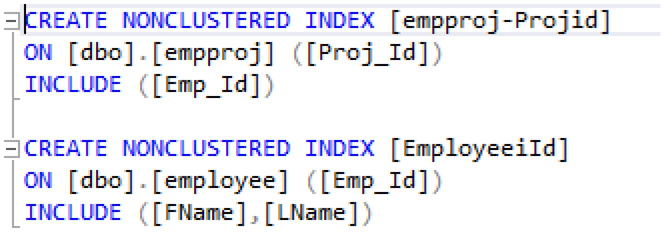
Query 3



```
Create PROC usp6
@porjName varchar(20)
AS
BEGIN
SELECT D.Dep_Name, D.Dep_Address
FROM Employee E , Department D, empproj em, Project P where
D.Manager_Id = E.Emp_Id  AND E.Emp_Id = em.Emp_Id AND em.Proj_Id = P.Proj_Id AND  P.Proj_Name !=@porjName
order by D.Dep_Name
END
```

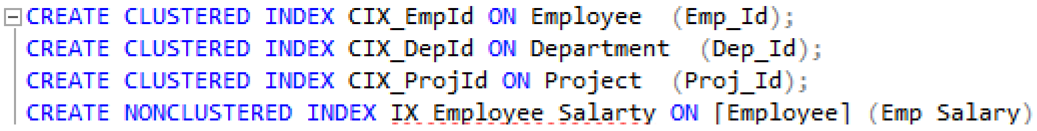
d) The modification in the indexes.

For query 1



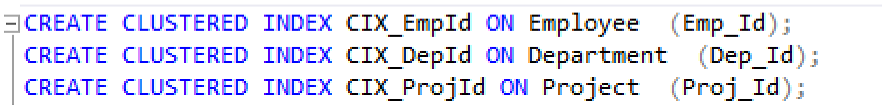
```
CREATE NONCLUSTERED INDEX [empproj-Projid]
ON [dbo].[empproj] ([Proj_Id])
INCLUDE ([Emp_Id])

CREATE NONCLUSTERED INDEX [EmployeeiId]
ON [dbo].[employee] ([Emp_Id])
INCLUDE ([FName],[LName])
```

For query 2

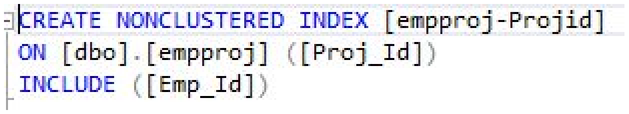


```
CREATE CLUSTERED INDEX CIX_EmpId ON Employee  (Emp_Id);
CREATE CLUSTERED INDEX CIX_DepId ON Department  (Dep_Id);
CREATE CLUSTERED INDEX CIX_ProjId ON Project  (Proj_Id);
CREATE NONCLUSTERED INDEX IX_Employee_Salarty ON [Employee] (Emp_Salary)
```

For query 3



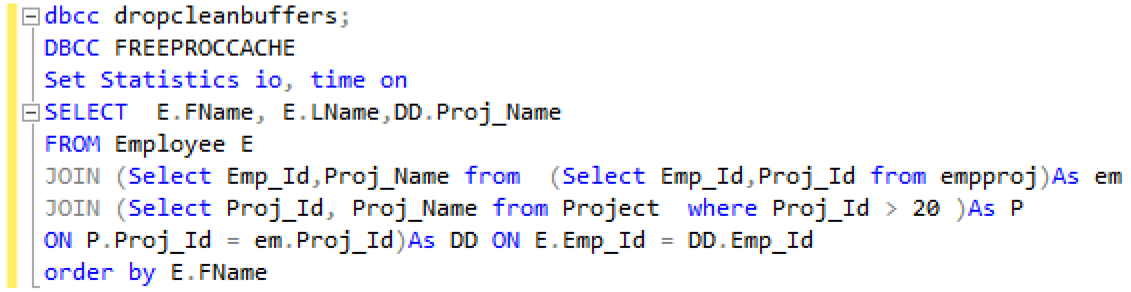
```
CREATE CLUSTERED INDEX CIX_EmpId ON Employee  (Emp_Id);
CREATE CLUSTERED INDEX CIX_DepId ON Department  (Dep_Id);
CREATE CLUSTERED INDEX CIX_ProjId ON Project  (Proj_Id);
```



```
CREATE NONCLUSTERED INDEX [empproj-Projid]
ON [dbo].[empproj] ([Proj_Id])
INCLUDE ([Emp_Id])
```

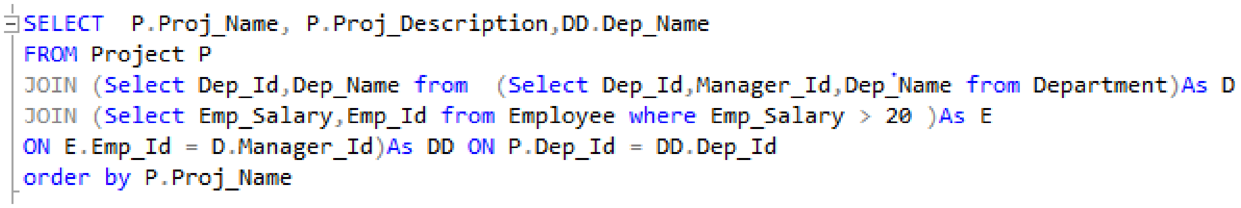
e. Modification in query system

Query 1



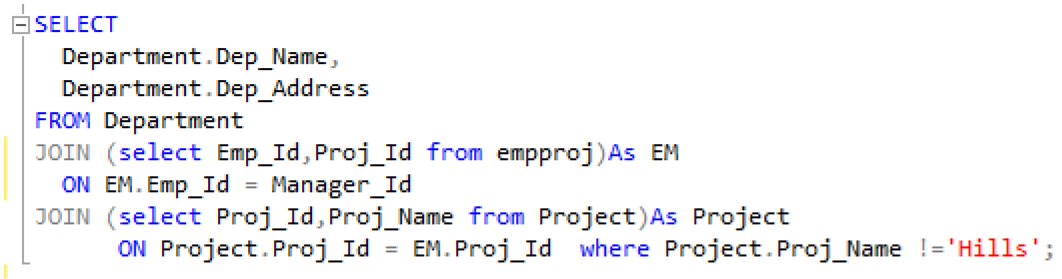
```
dbcc dropcleanbuffers;
DBCC FREEPROCCACHE
Set Statistics io, time on
SELECT  E.FName, E.LName,DD.Proj_Name
FROM Employee E
JOIN (Select Emp_Id,Proj_Name from  (Select Emp_Id,Proj_Id from empproj)As em
JOIN (Select Proj_Id, Proj_Name from Project  where Proj_Id > 20 )As P
ON P.Proj_Id = em.Proj_Id)As DD ON E.Emp_Id = DD.Emp_Id
order by E.FName
```

Query 2



```sql
SELECT  P.Proj_Name, P.Proj_Description,DD.Dep_Name
FROM Project P
JOIN (Select Dep_Id,Dep_Name from  (Select Dep_Id,Manager_Id,Dep_Name from Department)As D
JOIN (Select Emp_Salary,Emp_Id from Employee where Emp_Salary > 20 )As E
ON E.Emp_Id = D.Manager_Id)As DD ON P.Dep_Id = DD.Dep_Id
order by P.Proj_Name
```

Query 3



```sql
SELECT
  Department.Dep_Name,
  Department.Dep_Address
FROM Department
JOIN (select Emp_Id,Proj_Id from empproj)As EM
  ON EM.Emp_Id = Manager_Id
JOIN (select Proj_Id,Proj_Name from Project)As Project
      ON Project.Proj_Id = EM.Proj_Id  where Project.Proj_Name !='Hills';
```

Method 5:Optimization: Partitioning

- We made a partition for employee table to be employee table and empproj table.
- The empproj table contains the Emp_Id and Proj_Id that were in employee table.
- We remove Proj_Id from employee table.

3. Validation Details

a) After all optimizations on 1 million database

Query No.	No of rows	Time before millisecond	Time after millisecond	percentage of time enhancement.	NOSQL time
1	999984	21104	13401	36.5%	> 30 min
2	1000000	35413	8868	75%	> 30 min
3	999503	38701	9202	76%	> 30 min

Query No.	No of rows	Size of result rows before Bytes	Size of result rows after bytes	percentage of space enhancement
1	999984	35999424	35999344	2.22*10^-4
2	1000000	64 *10^6	64 *10^6	0
3	999503	39980120	39980120	0

After each Optimization

a. Index tuning

Query No.	No of rows	Time before millisecond	Time after millisecond	percentage of time enhancement.
1	999984	21104	15325	27.3%
2	1000000	35413	8744	75.3%
3	999503	38701	13491	65%

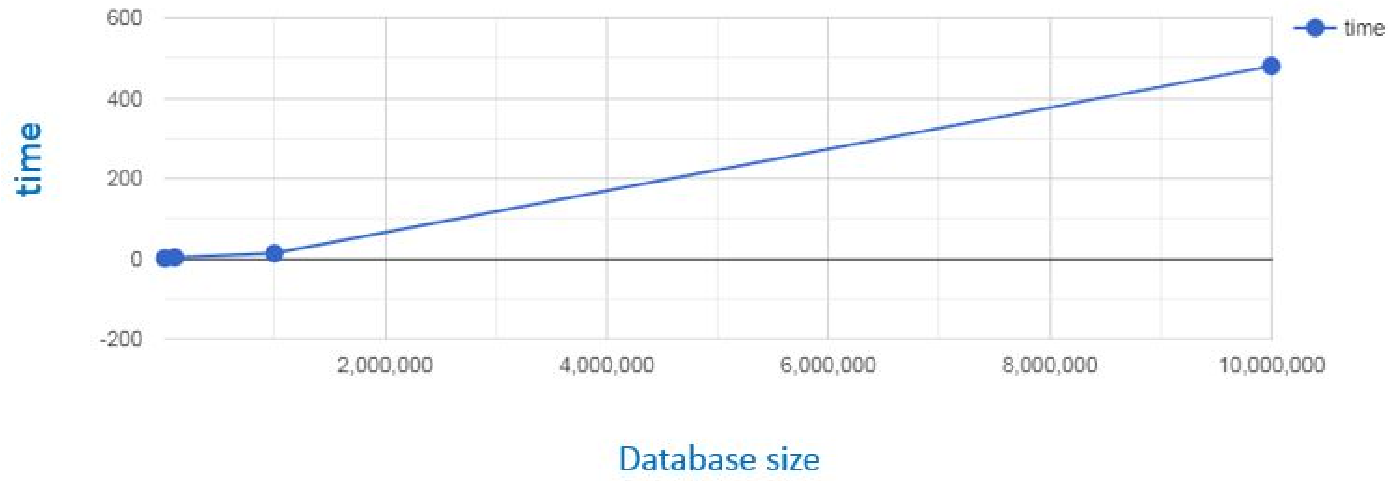
b. Query optimization

Query No.	No of rows	Time before millisecond	Time after millisecond	percentage of time enhancement.
1	999984	21104	14267	33%
2	1000000	35413	27293	23%
3	999503	38701	14206	63.2%

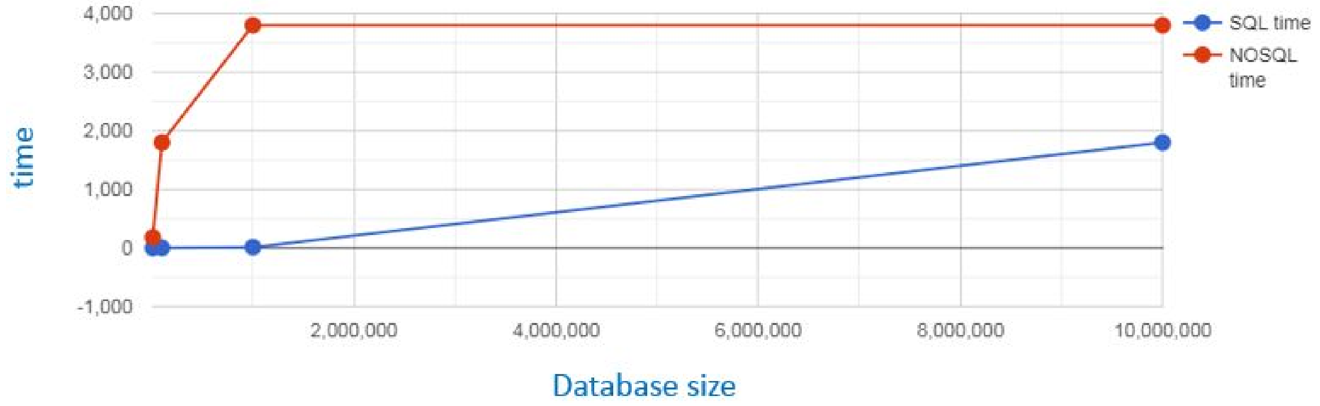
c. The enhancement in the memory management (stored procedures)

Query No.	No of rows	Time before millisecond	Time after millisecond	percentage of time enhancement.
1	999984	21104	10450	51%
2	1000000	35413	9031	75%
3	999503	38701	8217	79%

b) A graph explains the effect of the database size on performance (SQL Queries after optimization)



c) A graph explaining the different performance between SQL & NOSQL



Query No.		Time at database size = 10000000	Time at database size = 1000000	Time at database size = 100000	Time at database size = 10000
1	SQL	> 30 minutes	13.5 sec	2.5 s	0.5 sec.
	NO SQL	> 30 minutes	> 30 minutes	> 30 minutes	164.9 sec.
2	SQL	> 30 minutes	8.5 sec.	2 sec.	0.1 sec.
	NO SQL	> 30 minutes	> 30 minutes	> 30 minutes	181.8 sec.
3	SQL	> 8 minutes	10.5 sec.	2.5 sec.	0.5 sec.
	NO SQL	> 30 minutes	> 30 minutes	> 30 minutes	174.4 sec

NOSQL Queries

Query 1

```
db.project.aggregate([{
   $match:{
   Proj_Id: { $gt: 20 }
   }},{
     $lookup:{
       from: "empproj",
       localField: "Proj_Id",
       foreignField: "Proj_Id",
       as: "empprojs"
       }},{
   "$unwind": {
     "path": "$empprojs",
     "preserveNullAndEmptyArrays": true
```

```
    }}, {
    "$lookup": {
    "localField": "empprojs.Emp_Id",
    "from": "employee",
    "foreignField": "Emp_Id",
    "as": "empprojs.employee"
    }},{
   "$unwind": {
     "path": "$empprojs.employee",
     "preserveNullAndEmptyArrays": true
    }},{
    "$group": {
      "_id": "$_id",
      "Fname": {
      "$first": "$empprojs.employee.Emp_FName"
       },"Lname": {
       "$first": "$empprojs.employee.Emp_LName"
       },
       "Proj_Name": {
       "$first": "$Proj_Name"
      }}},{
        $sort: {
        Fname: 1
        }}
])
```

Query 2

```
db.employee.aggregate([{
   $match:{
   Emp_Salary:  { $gt: 20 }
   }},{
     $lookup:{
     from: "department",
       localField: "Emp_Id",
       foreignField: "Manager_Id",
         as: "departments"
       }},{
   "$unwind": {
     "path": "$departments",
     "preserveNullAndEmptyArrays": true
    }},{
    "$lookup": {
    "localField": "departments.Dep_Id",
    "from": "project",
    "foreignField": "Dep_Id",
    "as": "departments.project"
    }},{
```

```
   "$unwind": {
     "path": "$departments.project",
     "preserveNullAndEmptyArrays": true
    }},{
    "$group": {
      "_id": "$_id",
       "Proj_Name": {
       "$first": "$departments.project.Proj_Name"
       },
       "Proj_Description": {
       "$first": "$departments.project.Proj_Description"
      },
      "Dep_Name": {
       "$first": "$departments.Dep_Name"
      }}
      },{
        $sort: {
        Proj_Name: 1
        }}
])
```

Query 3

```
db.project.aggregate([{
   $match:{
   Proj_Name:  { $ne: 'Hills' }
   }},{
     $lookup:{
       from: "empproj",
       localField: "Proj_Id",
       foreignField: "Proj_Id",
         as: "emprojs"
       }},{
   "$unwind": {
     "path": "$emprojs",
     "preserveNullAndEmptyArrays": true
    }},{
    "$lookup": {
    "localField": "emprojs.Emp_Id",
    "from": "department",
    "foreignField": "Manager_Id",
    "as": "emprojs.department"
    }},{
   "$unwind": {
     "path": "$emprojs.department",
     "preserveNullAndEmptyArrays": true
    }},{
    "$group": {
      "_id": "$_id",
       "Dep_Name": {
       "$first": "$emprojs.department.Dep_Name"
       },
       "Dep_Address": {
       "$first": "$emprojs.department.Dep_Address"
      }}},{
        $sort: {
        Dep_Name: 1
        }}
])
```