

# Database Tuning, Optimization & performance analysis

## Introduction

In real world situations, creating database projects is more just making a scheme that fits the description. We are usually faced with a lot of questions like

- What DBMS should we use?
- Is it better to make it relational or NOSQL?
- How to get the best performance
- What Hardware spec. are recommended to use?
- ...etc

## Objectives

After this project, you should be able to

- Understand the concepts of database tuning.
- Get through the phases of database tuning.
- Use different database tuning techniques
  - Optimizing the schema
  - Optimizing the memory requirements
  - Optimizing queries.
- Use different database optimization tools.
- Practice Using NoSQL DataBases
- Perform analysis between using SQL & NOSQL Databases.
- And answer the above questions

## Project Phases

Phase#	Phase Name	Due date
Phase 1	Setup Database & fill it	2/1/2022
Phase 2	Optimization & Validation	16/1/2022

## Requirements

It is required to apply different database tuning methods on an existing database system. Students working in (3-4 students) team should pass the following project phases:

### **Phase 1: Setup database & fill it**

You are provided with an initial DB scheme (using MySQL), you need to understand it well, since you are going to modify it to enhance its performance.

You should fill your database with large volume of data (in the order of 1,000,000 to 10,000,000), you can use a database filling program which fills the database with random data. You will need to repeat this step after optimization so make sure you keep record of the generation steps.

You are preferred to use any free/open source DBMS like (MySQL, postgresql, ...etc)

#### **Deliverables of this phase**

1. Describe the existing system as well as hardware characteristics (the operating system, the CPU, the RAM, and the hard disk).
2. ER diagram for the system.
3. Database report describing the database statistics after filling (number of tables, max size of records in each table, number of rows, number of indexes, number of foreign keys, etc).

Example:

Table Name	Row Count	Main Key	Indexes	FK	Identity Column	Max Row Size (Bytes)
Category	16	Yes	3		Yes	59
Category_Keyword	409	Yes	2	2		8

Note: This report can be generated by the DBMS you use.

4. Schema implementation On NOSQL DATABASE

## **Phase 2: Optimization and Validation**

You will be provided with a set of queries, and required to run them once before optimization and once after optimization on relational DB. And only once on NoSQL DB.

### **Optimization**

In this phase you should use different optimization techniques to enhance your database performance, which reflect on the total system performance, in this phase you will pass through the following steps:

1. **Schema optimization:** your schema may need to be modified.
2. **Memory and cache optimization:** you can use stored procedures or any other techniques like changing the block size or any other parameter.
3. **Index tuning:** you can add or remove indexes as needed.
4. **Query optimization:** rewrite the query to enhance the performance.

**Note:** any modifications need to be justified.

### **Validation**

The final stage is to validate your optimization and analyze performance. For each step make sure to use the same laptop (N.B. Two different steps can be made on two different laptops)

- Run the queries on a large volume of data (in order of 100,000) on your database **before** and **after** optimization and compare the results.
- Run the queries on a large volume of data (in order of 100,000) **after** optimization and compare the results with NOSQL version.
- Finally report the effect of the **database size** on the system performance. This will be done by running the system on different database sizes (in the order of: 10,000 – 100,000 – 1,000,000 – 10,000,000) you can use, you can also try to change the hardware (run on another hardware device) and observe the change in the performance.

### **Deliverables of this phase**

A report describing the following:

1. Before and After Optimization Query Statistics
  - a. Execution plan for each query (Query tree)
  - b. Parallel query processing report in case of multiprocessor devices "Theoretically from the execution plan even it's not implemented by the DBMs".
2. Optimization Details
  - a. The new database statistics after modification "As in phase 1".
  - b. The enhancement in the schema.
  - c. The enhancement in the memory management.
  - d. The modification in the indexes.
  - e. For each query describe your modifications on the query statement.

3. Validation Details (These are Graphs and/or Tabularized Comparisons ONLY)
- a. For each query describe, For each enhancement:
    - i. The effect of all the previous modifications on the query performance, one enhancement at a time.
    - ii. The percentage of time enhancement.
    - iii. The percentage of space enhancement. (or degradation)
  - b. A graph explains the effect of the database size on performance ( Queries after optimization)
  - c. A graph explaining the different performance between SQL & NOSQL, (for fairness of comparison, write a small program to call the query on SQL, you can use a profiler as well)
  - d. The effect of changing the hardware specification. (if applicable)
  - e. Put your comments and recommendation for optimization, tuning & The choice of DBMS and recommend hardware (if applicable).