A. ABBAS

EVALUATING THE QUALITY ASPECTS OF SQL AND NOSQL DATABASES

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ABDULBASET ABBAS

A MASTER OF SCIENCE
THESIS
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

JUNE 2022

ATILIM UNIVERSITY 2022

EVALUATING THE QUALITY ASPECTS OF SQL AND NOSQL DATABASES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

ABDULBASET ABBAS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

JUNE 2022

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof. Dr. Ender
KESKİNKILIÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Computer Engineering, Atılım University**.

_____

Assoc. Prof. Dr. Gökhan
ŞENGÜL
Head of Department

This is to certify that we have read the thesis EVALUATING THE QUALITY ASPECTS OF SQL AND NOSQL DATABASES submitted by ABDULBASET ABBAS and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Damla Topallı
Supervisor

Examining Committee Members:

Asst. Prof. Dr. Cansu Çiğdem Ekin
Computer Eng. Department, Atılım University _____

Asst. Prof. Dr. Damla Topallı
Computer Eng. Department, Atılım University _____

Asst. Prof. Dr. Gül Tokdemir
Computer Eng. Department, Çankaya University _____

Date: *June 8, 2022*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Abdulbaset Abbas

Signature:

ABSTRACT

EVALUATING THE QUALITY ASPECTS OF SQL AND NOSQL DATABASES

Abbas, Abdulbaset

MSc., Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Damla Topalli

June 2022, 51 Pages

With the recent advances in technology and the growth of the data to be processed, choosing the suitable databases for your software has become a highly effective element that reflects directly on the overall quality and outcomes of any work. The most popular DBMS are either relational database management systems like SQL or non-relational database management systems like NoSQL; choosing between them at the early stages is essential. In this thesis, the key features of SQL and NoSQL databases have been discussed, considering the eight main software quality attributes affecting the database quality: Availability, Efficiency, Consistency, Durability, Maintainability, Reliability, Scalability, and Recovery Time. This study aims to understand the most effective quality attributes for SQL and NoSQL database implementation. Additionally, the popular DBMS tools for SQL and NoSQL implementation are discussed, considering four tools: MySQL, PostgreSQL, MongoDB, and Redis. To better understand the current implementation preferences, a questionnaire has been conducted with IT professionals: developers, database experts, testers, and managers to understand their preferences on SQL and NoSQL databases from the quality perspective. According to the questionnaire results, the highest important quality attributes for SQL are durability, consistency, and availability

whereas, scalability, durability, and efficiency were the most critical quality attributes for NoSQL.

Keywords: Relational databases, Non-relational databases, SQL, NoSQL, Quality Attributes

ÖZ

SQL VE NOSQL VERİTABANLARININ KALİTE YÖNÜNDEN DEĞERLENDİRİLMESİ

Abbas, Abdulbaset
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğr. Üyesi Damla Topallı

Haziran 2022, 51 Sayfa

Teknolojideki son gelişmeler ve işlenecek veri miktarındaki artış ile birlikte, yazılımınız için doğru veritabanlarını seçmek, genel olarak yazılımın kalitesine doğrudan yansıyan yüksek etkili bir unsur haline geldi. Günümüzde en çok kullanılan Veritabanı Yönetim Sistemleri (VTYS), SQL gibi ilişkisel veritabanı yönetim sistemleri veya NoSQL gibi ilişkisel olmayan veritabanı yönetim sistemleridir, projenin erken aşamalarında bu sistemler arasında doğru bir seçim yapmak çok önemlidir. Bu tezde, SQL ve NoSQL veritabanlarının temel özellikleri, Kullanılabilirlik, Verimlilik, Tutarlılık, Dayanıklılık, Sürdürülebilirlik, Güvenilirlik, Ölçeklenebilirlik ve Kurtarma Süresi olarak veritabanı kalitesi ile ilişkili sekiz ana yazılım kalite özniteliği dikkate alınarak tartışılmıştır. Bu çalışmanın temel amacı, SQL ve NoSQL veritabanı uygulaması için hangi özniteliklerin en etkili olduğunu belirlemektir. Ek olarak, SQL ve NoSQL uygulaması için popüler VTYS araçları tartışılmıştır: MySQL, PostgreSQL, MongoDB ve Redis. Ayrıca, mevcut uygulama tercihlerini daha iyi anlamak, kalite açısından sektörün SQL ve NoSQL veritabanlarındaki tercihlerini anlamak için geliştiriciler, veritabanı uzmanları, testçiler, yöneticiler gibi BT uzmanlarıyla bir anket yapılmıştır.

Yapılan anket çalışmasının sonucuna göre SQL veritabanları için en önemli kalite özniteliklerinin süreklilik, tutarlılık, ve kullanılabilirlik olduğu, NoSQL veritabanları içinse ölçeklenebilirlik, süreklilik, ve verimlilik özniteliklerinin, en önemli kalite öznitelikleri olduğu görülmüştür.

Anahtar Kelimeler: İlişkisel veritabanları, ilişkisel olmayan veritabanları, SQL, NoSQL, Kalite Öznitelikleri

To my parents and all my dearest siblings and friends

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my supervisor, Asst. Prof. Dr. Damla Topallı, for her assistance at every stage of the research project and for making a great effort to help and guide me in preparing the thesis with all her time and effort.

I would also like to thank examination committee members, Asst. Prof. Dr. Gül Tokdemir and Asst. Prof. Dr. Cansu Çiğdem Ekin for their valuable time, comments, and suggestions to improve this work.

Finally, I would like to express my gratitude to my parents and my family, who endured this long process with me, always offering support and love.

Abbas, Abdulbaset

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

DBMS	Database Management Systems
RDBMS	Relational Database Management Systems
SQL	Structured Query Languages
NoSQL	Not Only Structured Query Languages
API	Application Programming Interface
JSON	JavaScript Object Notation
BSON	Binary JSON
IBM	International Business Machine
ACID	Atomicity, Consistency, Isolation, Durability
CAP	Consistency, Availability, Partition Tolerance
BASE	Basically Available, Soft State, Eventual Consistency
ADR	Accelerated Database Recovery
IT	Information Technology
GUI	Graphical User Interface
DRBD	Distributed Replicated Block Device
GIN	Generalized Inverted Indexes
GiST	Generalized Search Tree
AGPL	Affero General Public License
REDIS	Remote Dictionary Server
POSIX	Portable Operating System Interface
API	Application Programming Interface
ANSI	American National Standards Institute

CHAPTER 1

INTRODUCTION

Recently, the amount of data to be processed is increased in real-world systems. To store and manage these data, databases play a critical role in the essence of information technology. A database is an organized series of data that stays accessed and managed without difficulty. These structures interact with various personal applications, various systems, and DBMSs for efficient data management. There are mainly two types of databases implemented: Relational DBMSs, also called SQL Databases, and Non-Relational DBMSs, also called NoSQL systems. The focus of the discussion in this thesis is to study how the software quality affects our choice between SQL and NoSQL databases. For this purpose, a questionnaire is conducted with IT professionals: developers, database experts, testers, and managers to understand their preferences on SQL and NoSQL databases from a quality perspective.

Relational databases are preferred when the application works with structured data in the form of related tables with a pre-defined structure. SQL, a standard programming language for database querying, is used to create, maintain and retrieve data and do transactions in relational databases. SQL databases ensure ACID (atomicity, durability, consistency, and isolation) properties for handling transactions. SQL databases are vertically scalable and can handle complex queries by using a standard SQL query language. It is an older approach compared to NoSQL, which developed in the 1970s. Because of this, there is highly available community support for SQL, which can be an important advantage since many experts can be found in this domain.

Considering the field of big data analytics, the database should be easily scale base on the need of the system. NoSQL databases are primarily preferred in such systems to handle huge amount of data. It is a better solution when the data changes over time, and not in a structured, tabular format. NoSQL systems are more flexible than the relational database systems, enabling high scalability and availability as the main

advantages of using these types of databases. Using cluster-based systems makes it easy to scale the NoSQL databases (horizontal scalability). It is also suitable for the hierarchical data storage. NoSQL systems follows CAP(consistency, availability, partition tolerance) theorem.

While designing real-world systems for managing and storing the data, the systems' performance and quality requirements should be considered. In addition, suitability to the software quality attributes is essential while developing such systems. In the literature, several studies reported the effect of some quality attributes on NoSQL databases, such as availability, consistency, durability, maintainability, performance, reliability, robustness, scalability, and stabilization and recovery time [1]. In another study by these authors, NoSQL literature is reviewed based on the given attributes: availability, scalability, durability, reliability, performance (read and write), and stabilization and recovery time [2].

Accordingly, in this thesis, the key features of SQL and NoSQL databases have been discussed, considering the eight main software quality attributes that is affecting the database quality: Availability, Efficiency, Consistency, Durability, Maintainability, Reliability, Scalability, and Recovery Time. This study aims to understand the most effective quality attributes for SQL and NoSQL database implementation. Additionally, the popular DBMS tools for SQL and NoSQL implementation are discussed, MySQL, PostgreSQL, MongoDB, and Redis.

This thesis is organized as follows: In Chapter 2, a background is provided to summarize the previous works in the literature related to the scope of this study. The methodology used for conducting the study is presented in Chapter 3. In Chapter 4, the results and discussion are given, and finally, in Chapter 5, the conclusion, limitations, and the future work of this study are presented.

CHAPTER 2

BACKGROUND OF THE STUDY

Several studies have been conducted in the literature to provide insight and report the effect of the advantages and disadvantages of the SQL and NoSQL databases. This chapter discusses SQL and NoSQL databases and their main features. Additionally, quality problems in the database system were discussed the main software quality attributes that significantly effects the database systems were identified.

2.1. SQL Databases and Features

The relational database model, also known as SQL databases, was first introduced by Codd (June 1970) and developed by IBM [3]. It is the traditional database model which is widely used for database management. In a relational database, the data is kept in tables, and the relations between them should be identified. These relations are formed using the unique identifiers known as the primary and foreign keys.

In these systems, a standardized programming language is defined, namely SQL. This language is used to create and manage the tables and retrieve the data. CRUD (create, read, update and delete) operations are performed, and stored procedures and views can be created using SQL. SQL is based on ACID (Atomicity, Consistency, Isolation, and Durability) properties when managing transactions. These properties are essential to keep the data consistent and preserve data integrity and reliability.

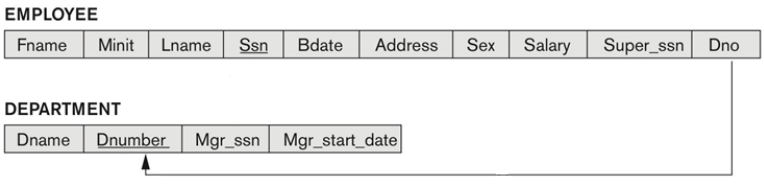

Figure 2. 1 An example of a Relational Database [4]

The relational schema for Company Database [4] is given in Figure 2.1, including two relations: Employee and Department. Employee holds information about the employees of a company, and Department identifies the information of departments (such as name and manager) that they worked in. The primary keys are Ssn and Dnumber, and unique identifiers for the Employee and Department relations, respectively. The attribute Dno of the Employee relation is a foreign key, as it references to Dnumber, the primary key of Department relation. It can be seen that in relational databases, the primary and foreign keys are used to conduct the relationships between two or more tables.

Traditional file access systems have several drawbacks, such as duplicate data and security issues for multiple accesses. Also, data consistency and integrity issues can be faced. Compared to traditional file systems, relational database systems provide more efficient performance when retrieving multiple records and in terms of indexing and key and redundancy control [5]. RDBMS provide certain functionalities that give them higher stability, performance, and consistency.

2.2. NoSQL Databases and Features

Considering the need for big data analytics in today's business environments due to the rapid data growth, processing data and retrieving performance by using SQL in relational databases becomes inefficient. To handle large amounts of data in such distributed and non-relational structures, cloud-based architectures should be used. However, SQL systems are not designed for cluster-based systems [6]. When considering scalability, flexibility with efficient performance when working on such systems, NoSQL systems are preferred. NoSQL (Not Only SQL) systems are also known as non-relational databases.

In contrast to relational database systems, these databases can scale up by adding new clusters (horizontal scaling) and distributing the data over those clusters. NoSQL provides easy data storage and retrieval, regardless of its structure and content, since there are no pre-defined data structures in non-relational databases [6]. To eliminate the limitations of relational database systems, NoSQL databases have emerged by providing high scalability, flexibility, and performance [7].

Several popular NoSQL database tools have been used in non-relational database systems. A literature review is conducted to have a detailed knowledge of these databases, such as Aerospike, Cassandra, Couchbase, CouchDB, HBase, MongoDB, and Voldemort, and introduced the quality attribute-based evaluation of NoSQL databases [1].

Four types of NoSQL database management systems use different data models for each: key-value store, document-based store, column-based store, and graph-based store.

Key-Value Store: The data is kept as keys and corresponding values using a hash table as seen in Figure 2.2. The data to be kept in this system can be text, images, or documents in a key-value format. This NoSQL database is easy to implement. Instead of complex queries, simply put and delete commands can be used [8].

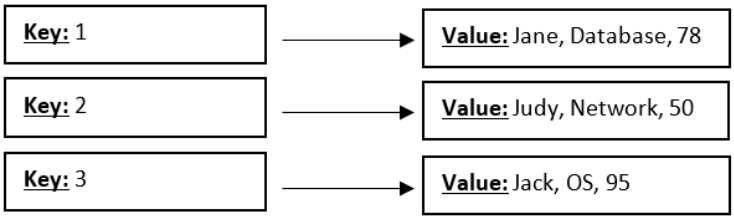

Figure 2. 2 Key-value model

Document-based Store: The records in this type of database are the documents. The data can be unstructured or semi-structured, where XML or JavaScript Object Notation (JSON) formats can be used; an example is provided in Figure 2.3. These are similar to the key-value type, with a unique identifier (primary key) for each document in the system. They are mainly used for web-based data, content management, and mobile applications.

```
{                                        {
"id":"b1"                                "id":"b2"
"name":"Fundamentals of Databases"       "name":"Game Design"
"genre":"Science"                        "genre":"Computer"
"price":"120.42"                         "price":"145.23"
}                                        }
```

Figure 2. 3 Example of Document-based Store

Column-based Store: The data is stored in column families. Each is associated with a row key, as shown in Figure 2.4. The group of related data accessed together is named a column family. Column-oriented databases are suitable for data mining and analytic applications [9].

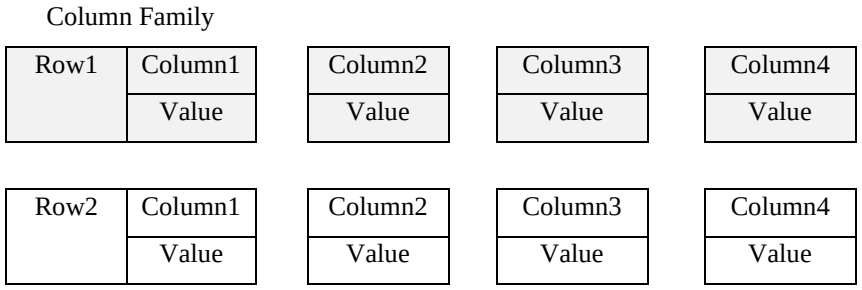

Figure 2. 4 Example for Column-based Store

Graph-based Store: This NoSQL database type is based on a connected network design, with the edges and nodes. The data is represented and stored in these edges and nodes as given in Figure 2.5. This type of database can be useful for Recommendation engines.

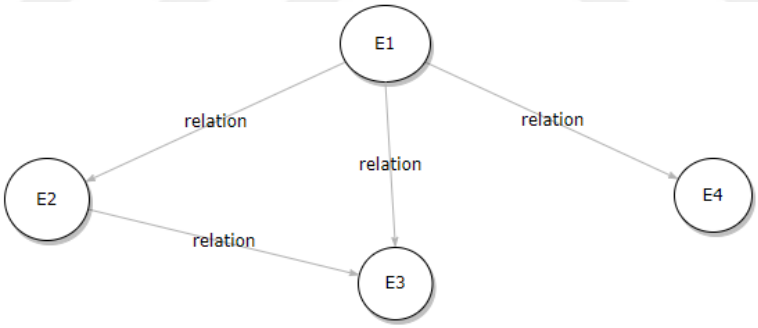

Figure 2. 5 Example for Graph-based Store

2.3. Addressing Quality Problems in Databases

The extent to which database quality issues impact the organization and the return on investment in data quality improvement is frequently used to help the business grow by introducing and developing a data quality management system.

Over the years, the companies faced many significant quality problems that affect the whole database system and reflect the overcome of the company and the workflow of the business, which leads in most cases to many defects in the outcome of the database system. The main problems can be listed as poor data organization, storing a massive amount of data, inconsistent data, poorly-defined data, incorrect data, data security, and recovery issues. These problems are described in the next section.

2.3.1 Poor Data Organization in Database

Good data organization is essential to conduct a successful research project. Data often has a longer lifespan than the related project since it can be reused for future projects or by other researchers. Data is essential for answering research questions, but much of it is lost or poorly managed. A lack of data organization can directly impact the project or future reuse. In this case, it is difficult to retrieve information if you cannot easily search through your data.

2.3.2 Too Much Data in Database

It is time-consuming to deal with a massive amount of data to get satisfactory results, especially if it is poorly organized. Due to the large amount of data to be stored, the database storage limits are also increased in the organizations. However, having too much poorly organized data may negatively impact the quality of the database.

2.3.3 Inconsistent Data in Database

If the data is obtained from several sources, such as different databases or data files, there can be an inconsistency problem for the records. Referential integrity constraints can be violated, which may cause problems for the data integrity. Another problem affecting the data quality is the duplicated data. When working with multiple data resources, identical records may be stored in the database more than once. This can also lead to inconsistency and reduce the data quality. As a solution, data cleaning methods should be applied, and the inconsistencies should be removed without losing the necessary information.

2.3.4 Poor Data Security

As organizations adopt database systems for day-to-day operations and decision-making, the security of data managed by these systems becomes crucial. Damage and misuse of data affect not only a single user or application but may cause serious problems affecting the entire organization [10].

The main aim of data security is to preserve the data and prevent data loss as a result of unauthorized access. The data should be protected against attacks that encrypt or destroy data or for unwanted modifications. Poor data security has a significantly destructive impact on the company's working process and causes untrustworthiness between the customers and companies.

2.3.5 Poorly Defined Data in Database

Poorly defined data means that data is defined in the wrong category. For example, the record, which can be changed over time, may lead to outdated information. Other problems may also arise due to the lack of data quality firewalls for critical data capture points or the lack of transparent cross-functional data processes.

2.3.6 Incorrect Data in Database

Usually, the presence of incorrect data leads to decreased condition-monitoring data quality in the database. Analyzing the poor-quality data will result in unreliable and inaccurate results. Many incorrect data occur because of the failures originating from the data acquisition equipment or data transmission and the diversity of data due to various formats[11], resulting in a significant decrease in data quality.

2.3.7 Poor Data Recovery From Database

Data recovery is one of the fundamental approaches when discussing database reliability. The failure in the recovery may result in the loss of vital information. Poor data recovery leads to losing a part or all the valuable data for the company or business; these types of losses can be harmful to the workflow of the business and negatively effects the quality of the database.

2.4. Software Quality Attributes and Trade-offs

Databases need to rely on high-level standards to ensure they can be trustworthy for businesses and clients. Several software quality attributes should be considered when developing a database, such as to measure the performance quality and ensure its continuity to deliver the appropriate services. In the following section, some of the leading quality attributes for databases are explained in detail, and how they can affect SQL and NoSQL databases differently is provided.

2.4.1 Availability

Availability means that any client making a data request should receive a response, even if one or more nodes are unavailable. In other words, a valid response should be provided to any request considering the working nodes in a distributed system. Since Relational database management systems (RDBMS) are mainly based on properties of ACID (atomicity, consistency, isolation, and durability). Due to this fact, these systems do not consider high availability [12]. As a solution, the concept of NoSQL came into existence considering the availability.

NoSQL databases work based on the CAP theorem principles (consistency, availability, and partition tolerance). In the literature, it is reported that many NoSQL databases have lost the demand for consistency to achieve better availability and partitioning [13]. Data replication is supported by most of the NoSQL databases for improved availability and data recovery. For data replication, multiple copies of data are kept on a number of clusters and data centers.

2.4.2 Efficiency

Efficiency is related to having the data easy to use, manage, and access. It is critical for large-scale businesses dealing with high numbers of records on immense networks. It is reported in a previous study that relational databases do not apply to clusters and efficiency problems were faced in these systems related to big data analytics [14]. If high efficiency is needed in an IT structure, NoSQL database can be suitable due to its distributed, cloud-based architecture [15], [16].

2.4.3 Consistency

In database systems, any given database transaction that modifies intended data appropriately is referred to as consistency. For the database to be consistent, data published to a database must be legal and conforming all defined rules, cascades, and constraints.

Consistency provides that there is no violation of database constraints due to programming faults. However, it does not ensure the accuracy of the database transactions. Since SQL offers a limited range of constraints to enforce logical integrity and SQL databases use ACID properties, they mostly satisfied strong consistency.

2.4.4 Durability

Durability for databases is the feature that guarantees transactions are preserved permanently and do not disappear or get deleted mistakenly, even during a database crash. The database must ensure the existence of its saved data even in extreme conditions that disrupt the data. In this context, all transactions should be stored in non-volatile data storage. In SQL, durability is preserved by keeping transaction logs in a system. On the other hand, NoSQL uses different methods to ensure high durability, such as: synchronously replicating writes across the three replicas or using data logging mechanisms.

2.4.5 Maintainability

Maintainability is how easily a system can be maintained or repaired by debugging or upgrading [17]. The defects should be corrected, and unexpected working conditions should be prevented to satisfy Maintainability requirements. In some cases, maintainability enables a continuous improvement system, and an enhancement is provided by learning from past experiences of maintenance.

In SQL systems, database administrators have the capability to easily control and update the data in the databases, providing high maintainability. Automation tools of RBDMS are used to do the tasks like taking back-ups of the data in an automated

manner. However, it is reported that NoSQL systems offer limited maintainability when compared with traditional RDBMS [18].

2.4.6 Reliability

The databases' ability to run consistently without errors is called reliability, and it ensures that data is accurate and consistent. Data reliability is highly related to data safety, recoverability, and integrity.

For instance, if a database system is not reliable, inconsistent, or accurate, this affects the results of data analysis, and problems in decision-making occur. Since the database is a critical component of organizational infrastructure, its failure can cause significant problems throughout the business. Hence, database reliability is vital for organizations.

High reliability is obtained for the database transactions in traditional RDBMS and SQL systems since these systems support the ACID properties [19]. However, there is a lack of reliability in NoSQL systems since these databases provide eventual consistency as a principle of BASE (Basically Available, Soft State, Eventual consistency) properties [20].

2.4.7 Scalability

The ability to extend the capacity of system resources to support your applications is known as scalability. There are two types of scaling: vertical and horizontal scaling. Vertical scaling is the process of accelerating the processing power of a single server or cluster. The scaling-up process can be performed for relational and non-relational database systems up to the maximum limit of a throughput. Additionally, using high-performance systems (servers, clusters, etc.) may increase the expenses. Figure 2.6 illustrates the Vertical scaling as given in below.

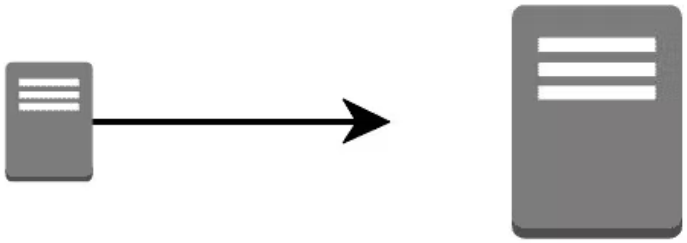

Figure 2. 6 Vertical Scaling

Most SQL databases are vertically scalable. In these systems, the load on a server or cluster can be increased by improving the memory and processor capabilities of the system.

The second type is horizontal scaling, where the workload is distributed among different nodes. This approach may cause problems when applied in RDBMS systems since managing the related data across distributed nodes is challenging. On the other hand, for non-relational NoSQL databases, horizontal scaling is appropriate since the data are self-contained, no relations are determined, and there is no need for join operations. There is no restriction to keep the data among distributed nodes.

Figure 2. 7 Horizontal Scaling

In NoSQL database implementation, the increased workload can be handled by adding more servers, and nodes to the system as they are horizontally scalable.

2.4.8 Recovery Time

Recovery time is needed to restore the system to a normal state. Any disruption to database services, whether caused by a natural catastrophe, equipment failure, cyber assaults, or other factors, can prevent an organization from executing normal

procedures and conducting daily business. This may result in dissatisfied customers, lost income, and a ruined reputation. A company must have a disaster recovery policy in place to minimize disruptions to database services, mainly when serving mission-critical workloads.

SQL databases provide multiple tools to ensure stable recovery time, such as ADR (Accelerated Database Recovery) introduced by SQL Server. While for NoSQL databases, the resilient architecture of NoSQL databases typically provides a buffer against data loss, MongoDB and Aerospike are good picks if looking for good recovery time[2].

2.5 Database Management Systems: Popularity Rankings

While studying databases systems in depth, it's important to consider their popularity side by side to their beneficial features to your business, to be able to take an early decision to which of those systems are highly considerable choices, then apply more detailed filters that take you steps further to choose the best database system the fulfill your goals.

According to DB-Engines[21], the top 10 popular database systems between May 2021 and May 2022 is given in Table 2.1.

Table 2.1 Top 10 DBMS Popularity Ranking 2021-2022 by DB-Engines [21].

DBMS	Database Model	May 2022 Ranking	May 2021 Ranking	May 2022 Popularity Score
Oracle	SQL	1	1	1262.82
MySQL	SQL	2	2	1202.10
Microsoft SQL Server	SQL	3	3	941.20
PostgreSQL	SQL	4	4	615.29
MongoDB	NoSQL	5	5	478.24
Redis	NoSQL	6	7	179.02

IBM Db2	SQL	7	6	160.32
Elasticsearch	NoSQL	8	8	157.69
Microsoft Access	SQL	9	10	143.44
SQLite	SQL	10	9	134.73

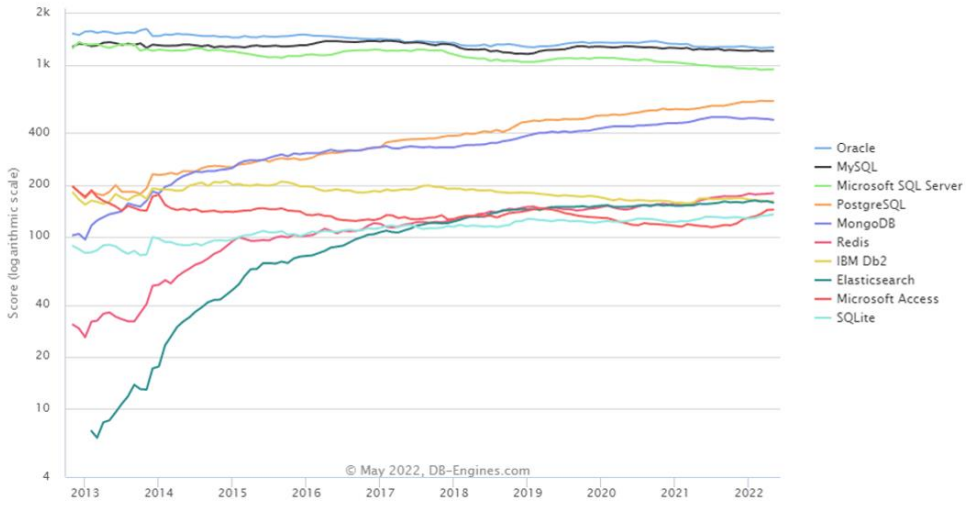

Figure 2. 8 Popularity Score Ranking of Top 10 DBMSs [21].

As seen in Table 2.1 and Figure 2.8, there were 7 SQL database management tools in this list, whereas 3 NoSQL tools were found. The lead of the list is “Oracle” it was the first business to launch an RDBMS platform in 1979, and it is still the market leader in terms of revenue. Like other RDBMS software, Oracle Database is built on SQL, a standardized programming language used by database administrators, data analysts, and other IT professionals to administer databases and query the data stored in them.

MySQL is the second-ranked tool in the list, which is a flexible and easy-to-use RDBMS in addition to its ability to modify the source code to meet your expectations.

Microsoft SQL Server is in the third rank. It is RDBMS, developed by Microsoft. The main component of the SQL Server is called “Database Engine”. It actually can support thousands of users simultaneously. Then, PostgreSQL, MongoDB and Redis tools were listed.

IBM DB2 is in the seventh rank, which is a collection of products for database and data management, processing, and analytics. The relational database concept is firstly mentioned by Codd, in IBM [3]. SQL, the standard database language, was also created by IBM. In 1983, they dubbed their database product DB2 [22].

Elasticsearch is a distributed full-text NoSQL database that is based on Apache Lucene and was first released in 2010. It utilizes documents rather than schema or tables. It is also a free and open-source program that allows for real-time data search and analysis. Elasticsearch was created to store, retrieve, and manage structured, unstructured, and semi-structured data. It is well-known for its ability to provide direct, easy, and quick access to stored data and its search engine's scalability due to its distributed architecture.

MS Access is coming up next, which is developed by Microsoft. MS Access includes a user-friendly GUI and application development tools with the relational Microsoft Jet Database Engine [23]. It is appropriate for both small and big database implementations, and this is due in part to its simple graphical interface and its compatibility with various apps and platforms.

Finally, SQLite is on the tenth rank, also an RDBMS, and is an embedded SQL database engine, not using a client-server architecture. Unlike other DBMS tools, the database in SQLite is kept in a single file. This fact provides ease of accessibility. For instance, copying a database is as simple as copying the file that stores the data. Cross-platform between systems is also supported.

According to these findings, an in-depth comparison is made between four of the top ten popular DBMS: two DBMS tools, MySQL and PostgreSQL, and two NoSQL DBMS tools, MongoDB and Redis, are discussed in the next section.

2.5.1 SQL DBMS Comparison (MySQL - PostgreSQL)

First, the information about MySQL is provided, which is an RDBMS works based on the relational database model, to make database performance more quickly and flexible. The SQL standard used in MySQL commonly means the current version of the SQL Standard at any time [24]. As seen in Figure 2.9, MySQL has much higher popularity than PostgreSQL, which has been almost at a constant rate for ten years.

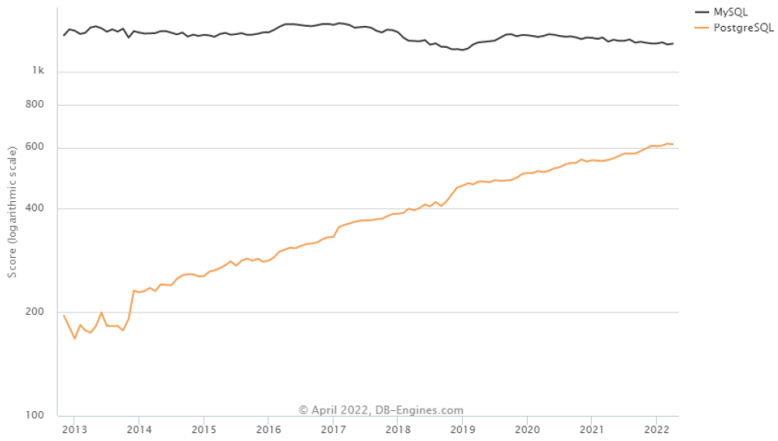

Figure 2. 9 MySQL vs PostgreSQL Popularity Ranking [21].

As a result, for massive amounts of data contained in large data warehouses and Big Data applications, MySQL enables high scalability to coordinate and manage the data. Another essential feature provided by MySQL is flexibility.

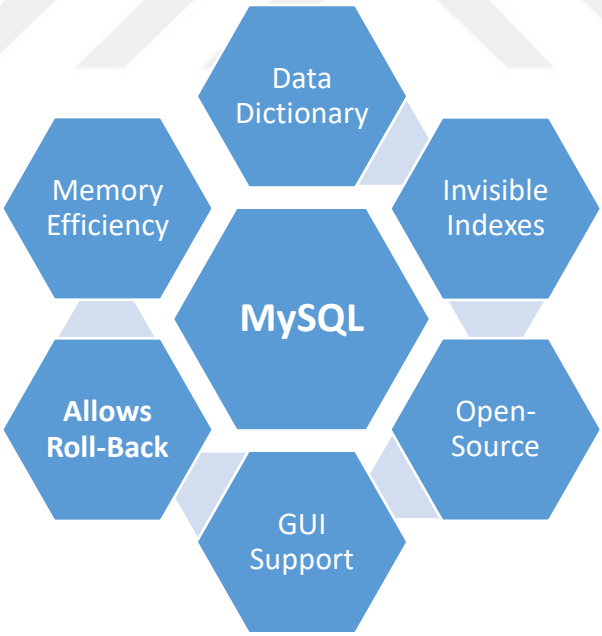

Figure 2. 10 MySQL Key Features

MySQL is mainly used considering the cases below:

- Explicit support for memory storage engine for frequently used tables.
- To have scale-out features.
- A simple, fast, and read-heavy workflow RDBMS.
- A vast number of 3rd party tools are available for developers and database administrators.
- An RDBMS provides a roll-back feature that allows transactions to be rolled back and crash recovery.

PostgreSQL is a highly stable DBMS and an open-source database supporting relational SQL and non-relational queries by JSON. More than 20 years of community development it has made a significant contribution in terms of integrity, resilience, and correctness [25]. It is highly fault-tolerant due to the write-ahead logging facility. It has compliance with ACID properties, supporting primary key- foreign keys, joins, views, and stored procedures. The standard SQL is used as the interface language of PostgreSQL [26].

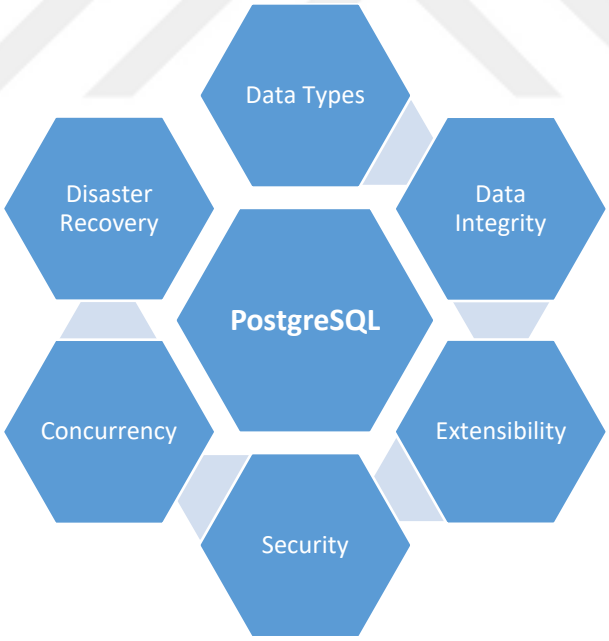

Figure 2. 11 PostgreSQL Key Features.

PostgreSQL is widely preferred when:

- An RDBMS is designed with standards compliance, scalability, and data integrity.

- A free and open-source RDBMS.
- Both users and roles can assign object-level privileges.
- To use advanced data types such as arrays or customed user-defined types.
- A supports triggers on a large variety of commands in the database.

2.5.2 NoSQL DBMS Comparison (MongoDB - Redis)

Considering NoSQL DBMS, as seen in Figure 2.12 one of the most popular NoSQL tool is MongoDB, an open-source project with an AGPL license held by the 10gen company, developed in 2009. It is a schema-free and document-oriented DBMS tool, storing the data in JSON or BSON format. It also allows the integration of data with certain types of applications and provides horizontal scalability. Hence, it is highly scalable. Different from other tools like Cassandra, a MongoDB cluster includes an arbiter, a master node, and multiple slave nodes [27].

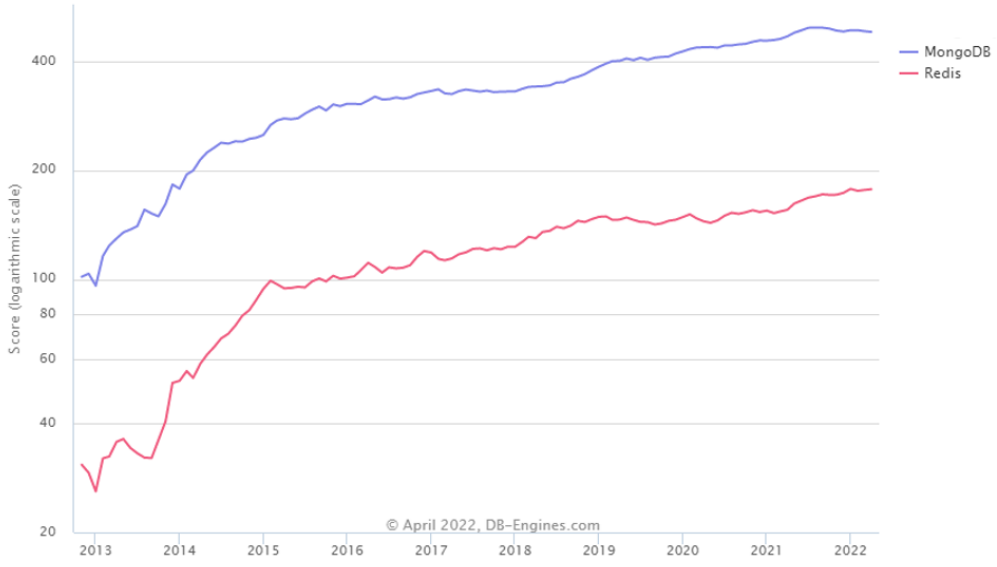

Figure 2. 12 MongoDB vs Redis Popularity Ranking [21].

As with all NoSQL systems, in MongoDB, there are no schema restrictions and can support semi-structured data like XML, JSON, and CSV and multi-attribute lookups

on records that may have different kinds of key-value pairs [28]. The data can be stored in two ways: as nesting documents inside each other or as references to the other documents.

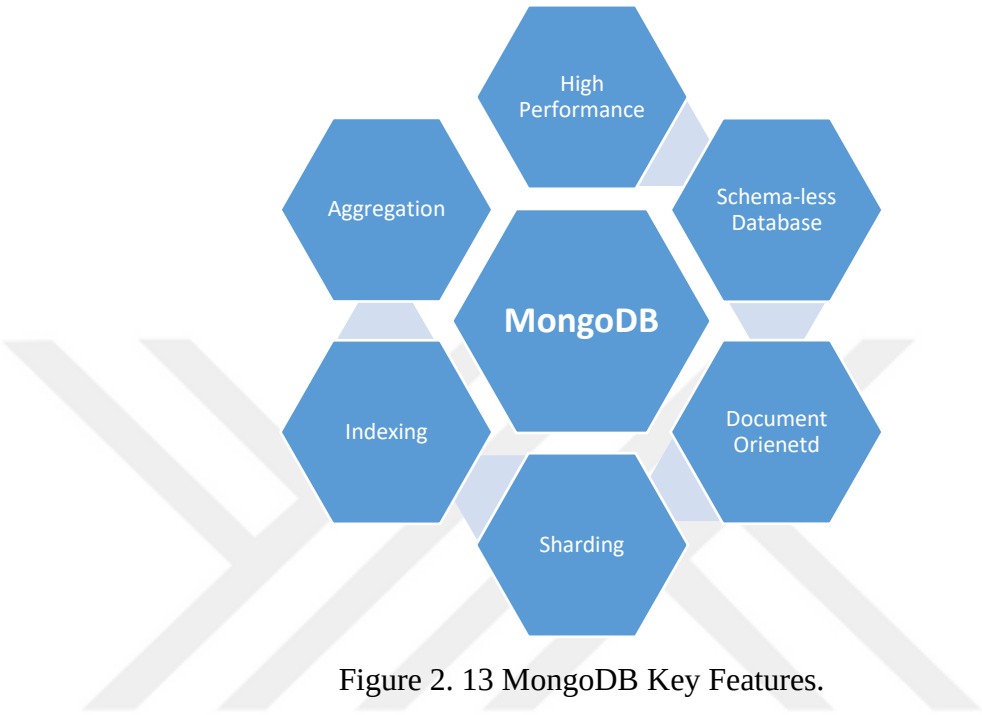

Figure 2. 13 MongoDB Key Features.

MongoDB is preferred when:

- High speed and stable performance system is needed
- Great scalability with the ability to support the horizontal scaling of data.
- To ensure high availability by using data replication features where copies of data are kept on several data servers.
- To store or manipulate massive amounts of data.

Finally, the second NoSQL tool to discuss is Redis. Redis or "Remote Dictionary Server" is an open-source, networked, single-threaded, in- memory, advanced key-value store with optional durability. It is also a well-known in-memory data platform that may be used as a cache, message broker, and database and can be installed on-premises, in the cloud or in a hybrid environment.

Redis is built-in ANSI C and is compatible with POSIX platforms such as Linux, Mac OS X, and Solaris. It is accessible to apps via its client API library. Redis, which offers a reasonably vast collection of data types compared to many key-value data stores, maintains its database entirely in memory, with a disk used exclusively for durability. The read and write performance is good even the workloads increase. The data is mostly held in memory and written to disk asynchronously from time to time to achieve persistent storage. Because of this in-memory nature, the complexity is less, and the application boasts exceptional performance for the read and write operations [29].

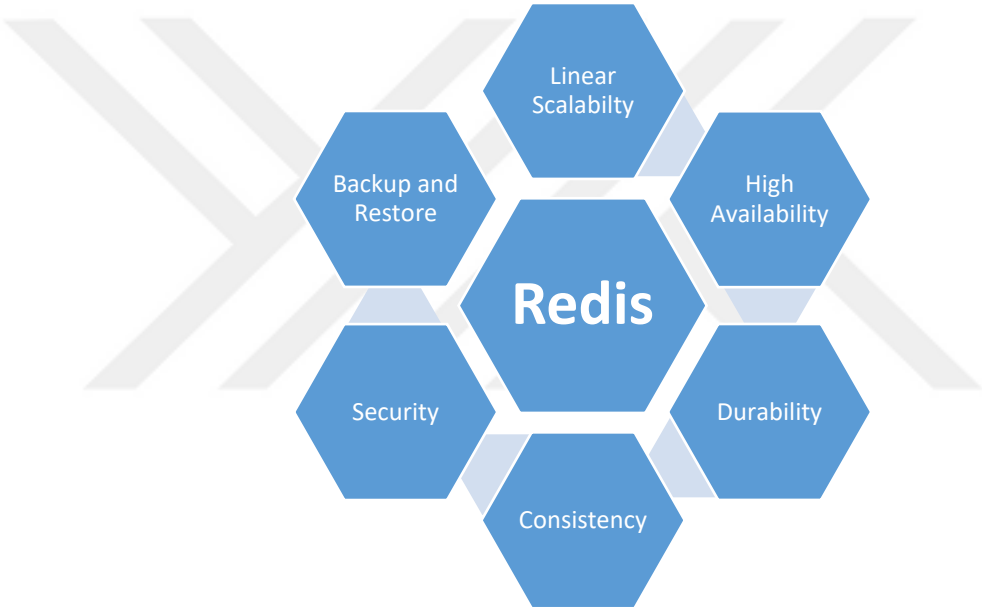

Figure 2. 14 Redis Key Features.

Redis is preferred for:

- High-speed DBMS, Redis is extremely fast compared to other datastores because it's an in-memory database.
- DBMS that uses Key-Value stores database structure.
- An in-memory data structure platform that allows caching and supporting message brokers [30].

CHAPTER 3

METHODOLOGY

In this study, our main aim is to understand the key features, and differences between SQL and NoSQL systems, and to evaluate the importance of the quality attributes in each of these systems. In this context, a questionnaire is conducted with IT professionals including developers, database experts, testers, and managers to understand their preferences on SQL and NoSQL databases from their quality perspective. Thus, this study is conducted to understand the effects of the eight software quality attributes related to the databases, differently for SQL and NoSQL databases, as well as to simplify the choice of which kind of DBMS the IT professionals prefer for their business according to the quality factors which should be considered the most.

3.1 Research Questions

In this study, it is aimed to answer the following main two research questions:
RQ1. What are the key features of SQL and NoSQL databases?
RQ2. Which of the quality aspects (Availability, Efficiency, Consistency, Durability, Maintainability, Reliability, Scalability, Recovery Time) are essential in SQL and NoSQL?

3.2 Research Procedure

To answer the research question RQ1, a literature review was conducted. For RQ2, an online questionnaire was performed. As making further steps to identify the most suitable DBMS for your company or business, a questionnaire has been made using Google Forms and LimeSurvey and then published between the time scope of 8/1/2022 and 18/5/2022, the survey contained a total of 13 questions divided between questions related to the demographics to be able to understand the type of participants who

participated voluntarily in the survey, as well as their knowledge and scope of experience, while the other parts were focused on the eight software quality attributes and how the participants evaluate their effect differently for SQL databases and NoSQL databases, in addition to the participant opinion to which of the software quality attributes do you think are important for relational and non-relational databases, whereas the last part was an open question where the participant able to add any important information regarding this topic. The survey has been distributed by using online channels to a broad range of groups, including university students, professionals and managers working in IT companies, to obtain a different wide range of opinions which have been derivative from different sides of theoretical and practical information.

3.3 Data Sources

To answer RQ1, all the major databases are searched to gain a broad perspective of the related work in the literature. Database searching is more beneficial than searching through a limited set of journals and conference proceedings. Table 3.1 provides the list of the four main databases that were explored in this study.

The reason for selecting these databases was the inclusion of the relevant journals, workshop proceedings, conferences, and symposiums within the domain of key-value stores. The work in progress, workshop reports, and technical reports were excluded. The duplicated studies obtained from different resources were also excluded through manual screening.

Table.3.1 Data Sources.

Data Origins	**Web Sites**
IEEE Xplore	http://ieeexplore.ieee.org/Xplore/
Scopus	https://www.scopus.com/
Web of Science	https://www.webofscience.com/
Google Scholar	https://scholar.google.com/

3.4 Instruments

In this thesis, a questionnaire is prepared including 13 questions (see Appendix A for the original questionnaire in Turkish, and Appendix B for English version of the questionnaire). The last question is an open-ended question that the respondents can share their experiences or preferences regarding relational and non-relational databases. When preparing the questionnaire, the items were decided and used after gathering the views of three domain experts in the Software Engineering field, and necessary corrections were made for the content validity.

There was a total of 16 choices for the eight quality attributes considering both SQL and NoSQL databases. The importance of each quality attribute is evaluated by the scale 5-point Likert-type with the following level of agreement: Strongly Disagree (SD), Disagree (D), Neutral (N), Agree (A), Strongly Agree (SA). To test the validity, the Cronbach Alpha coefficient was calculated as 0.875 for internal reliability, showing a high degree of internal reliability. In testing the hypothesis of the research, the .05 significance level was utilized.

3.5 Participants

In this survey, a total of 59 attempts made, but only full responses were considered for the analysis (the partial responses were omitted). There were 45 full responses obtained through the questionnaire. The details of the demographics of the participants were provided in below.

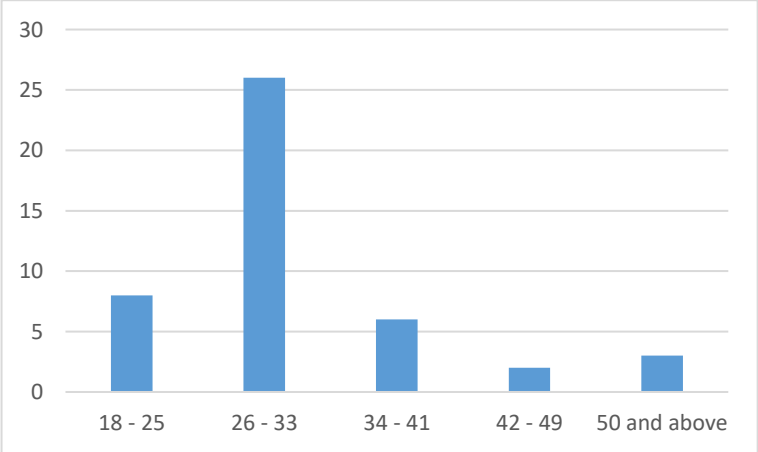

Figure 3. 1 Participants' Age Distribution

As seen in Figure 3.1, considering the participants' age group distribution, most of them were between 26-33 years (26 participants, 58%), then 18-25 (8 participants, 18%), and 34-41 age group (6 participants, 13%). However, the senior group is very low where only 2 participants from the age group 42-49 (4%) and 2 participants from 50 and above (7%).

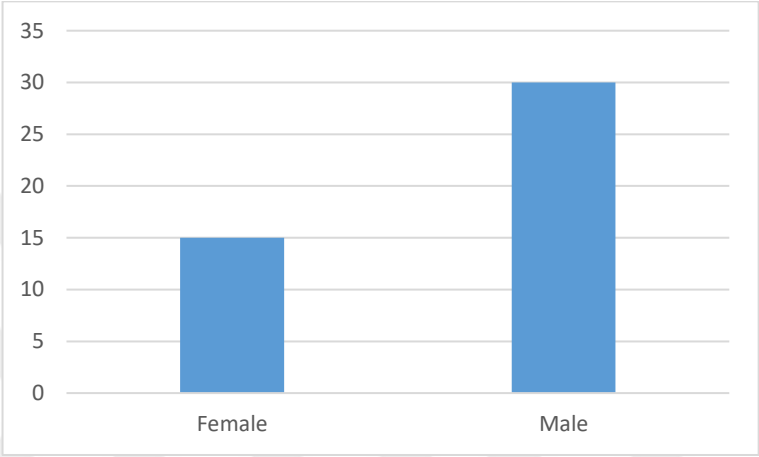

Figure 3. 2 Participants' Gender Distribution

There were 15 female participants (33%), whereas 30 male participants (67%) filled out the questionnaire (see Figure 3.2).

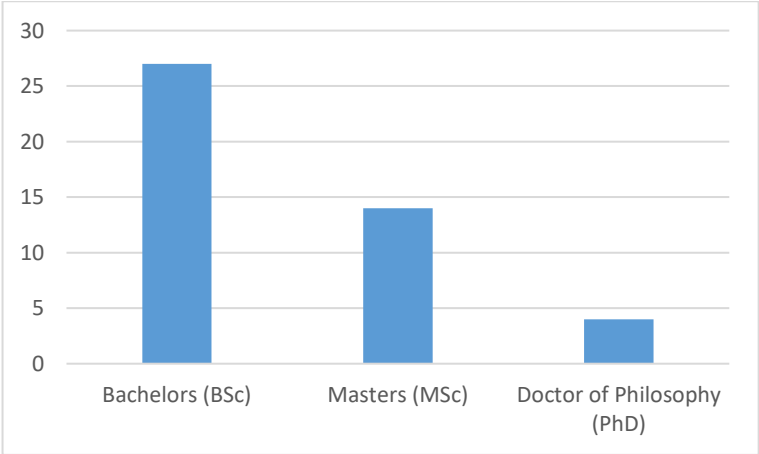

Figure 3. 3 Diploma Degrees of Participants

As seen in Figure 3.3, the majority of the participants have a diploma degree of Bachelors (BSc) (27 participants, 60%), after that, 14 participants completed the Masters (MSc) degree with 31%. Very few Ph. D graduates, only 4 (9%) participated in this study.

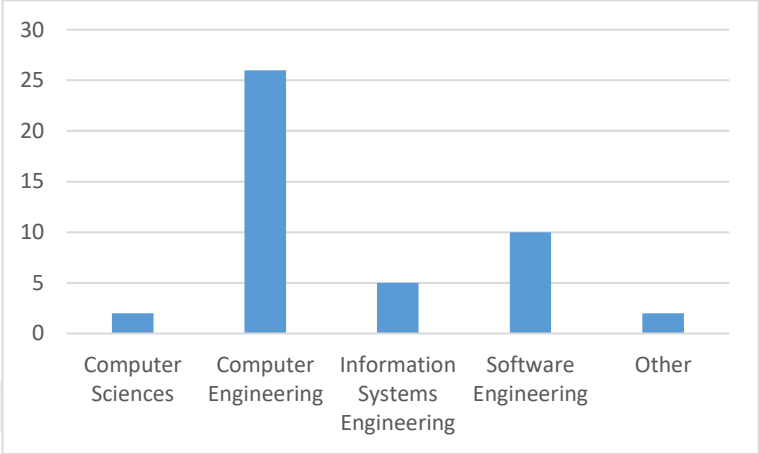

Figure 3. 4 Department of Participants'

Figure 3.4 depicts the departments of the participants; the majority of them graduated from the Computer Engineering Department (26 participants, 58%). Ten were from Software Engineering (22%) and five from Information Systems Engineering (11%). There were two participants from the Computer Science (4%) and two coded as Other (4%): one graduated from Mathematics and the other from the Electrical and Electronical Engineering department.

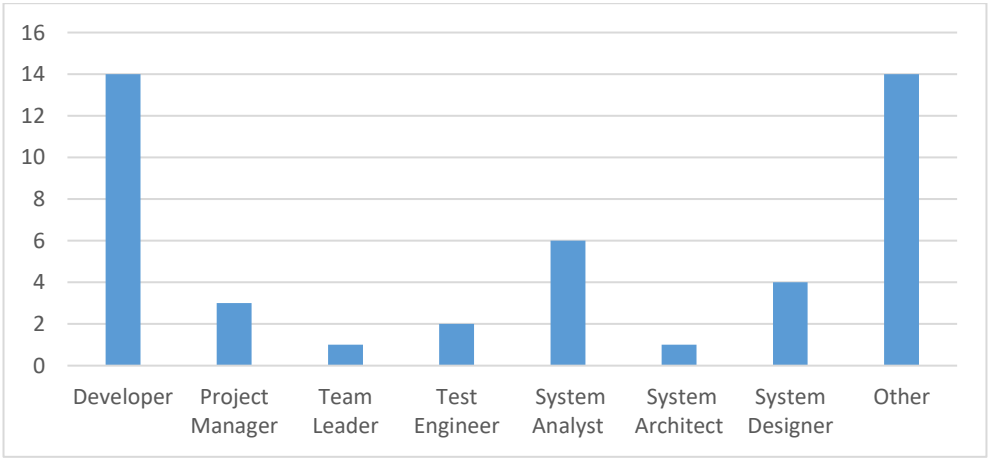

Figure 3. 5 Distribution of Occupation

As seen in Figure 3.5, most of the participants were Developers (14 participants, 31%) and System Analysts (6 participants, 13%). There was only one team leader (2%) and three project managers (7%) since the number of experienced senior participants were very limited. Others include Data Analysis and Analytics, Data Management consultants, Database Specialists, Software Support Specialists, Security Specialists, and Research Assistants (31% in total). Also, there were two Test Engineers (4%) and four System Designers (9%).

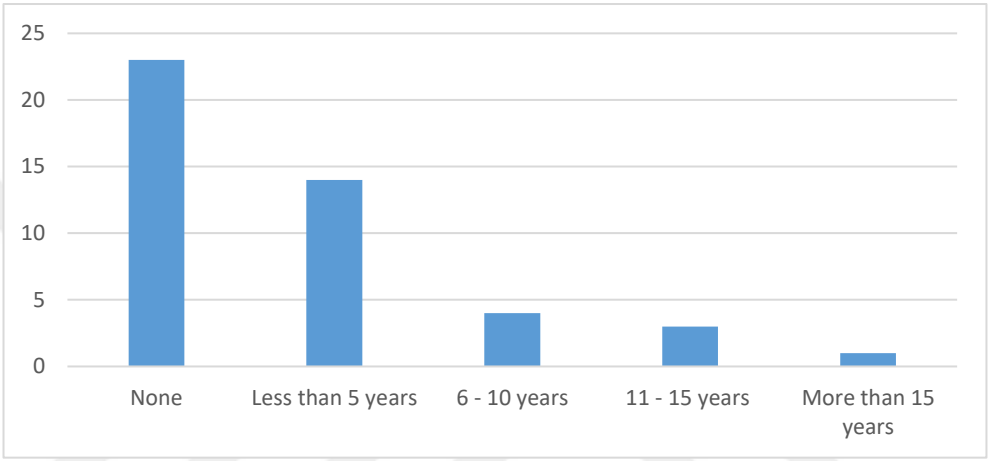

Figure 3. 6 Project Managers' Information

Considering the participants' experience, the responses for the question regarding the project managers are given in Figure 3.6. Similar to the previous result, the number of experienced participants acting as project managers for 11-15 years or more is only 4 (9%). Twenty-three participants (51%) claimed they did not work as project managers.

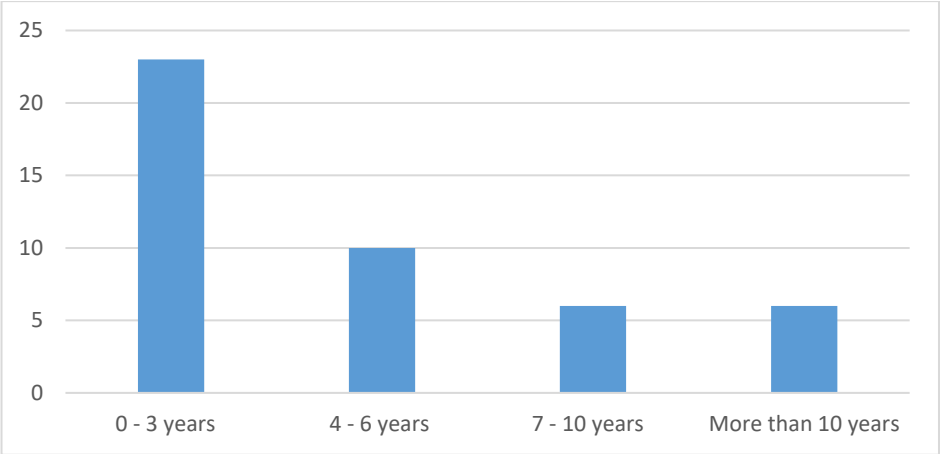

Figure 3. 7 Years of Work Distribution

Finally, the years of work information is presented in Figure 3.7. The participant's professions were mostly between 0-3 years with a total of 23 participants (51%), then we have relative values to the experience around 4-6 years and 7-10 years with a total of 16 participants for both of them (35%). The remaining six participants have worked for more than ten years (13%).

CHAPTER 4

RESULTS and DISCUSSION

The questionnaire responses are analyzed descriptively, and the results were discussed in this section. Firstly, the effect of the software quality attributes for SQL databases is given in Table 4.1.

Table 4.1 SQL and Quality Attributes

Quality Attribute	**Disagree**		**Neither agree nor disagree**		**Agree**			
	N	**%**	**N**	**%**	**N**	**%**	**Mean (M)**	**Standard Deviation (SD)**
Consistency	3	5.08%	10	16.95%	34	57.63%	3.94	0.94
Availability	2	3.39%	13	22.03%	32	54.24%	3.89	0.91
Durability	5	8.47%	7	11.86%	35	59.32%	3.85	0.96
Efficiency	1	1.69%	14	23.73%	32	54.24%	3.85	0.83
Reliability	5	8.47%	14	23.73%	28	47.46%	3.77	1.16
Maintainability	8	13.56%	13	22.03%	26	44.07%	3.57	1.06
Scalability	12	20.34%	16	27.12%	19	32.20%	3.19	1.24
Recovery Time	14	23.73%	16	27.12%	17	28.81%	3.04	1.06

Table 4.1 and Figure 4.1 show that consistency has the highest mean score for SQL databases with 3.94. 34 respondents (57.63%) agreed that consistency is important for SQL databases. Additionally, most participants (59.32%) agreed that durability is the most effective quality attribute for relational databases. It is an expected result since ACID concept comes from the SQL environment [31] and relational databases support atomic transactions. Durability is a very effective quality attribute for every SQL databases.

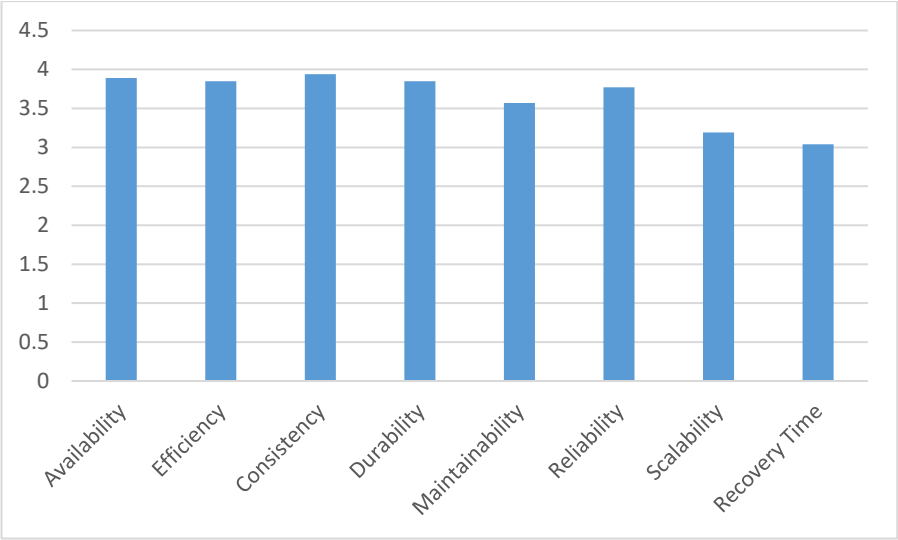

Figure 4. 1 SQL and Quality Attributes Mean Scores.

Considering mean scores, the other two essential quality attributes can be reported as availability with 3.89 mean and efficiency with 3.85 mean, respectively. 54.24% of the respondents agreed that availability and efficiency is important for SQL databases. Even though availability is necessary for all database systems, RDBMS does not quite fit high availability. Hence to satisfy these needs, the concept of NoSQL came into existence [12]. While for efficiency for RDBMS, in many cases, database performance issues are caused by inefficient SQL queries, they also have trouble with efficiently expanding a database due to their inherent complexity of organization [32]. To the participants' opinion about the high effect for efficiency on SQL databases, it is possible to enhance the efficiency for RDBMS by optimizing queries by using a database performance analysis solution to ensure high efficiency.

Reliability got a total agree of 28 votes, and 3.77 mean score, because SQL databases are based on ACID properties, so they have a good and stable reliability rate while keeping the integrity of data. After that, maintainability with a total of 26 agrees, and a 3.57 mean is reported. Scalability got a fairly low number of votes with 19 agree votes and 3.19 mean. In general, RDBMS databases are vertically scalable, so when the load on RDBMS database, we have to scale the database by increasing server hardware power and need to buy expensive and large servers. This was always considered the main problem for SQL databases and caused companies with vast

amounts of data not to prefer to use them. And the least number of votes was for recovery time with total agree votes of 17 and 3.04 mean. Even though recovery time is considered less important, it should be noted that the data in relational databases can be corrupted due to hardware issues, software bugs, and user errors [33]. SQL provides multiple tools to ensure stable recovery time, such as ADR (Accelerated Database Recovery) introduced by SQL Server.

While for NoSQL databases, the same question was asked about the effect of the software quality attributes of NoSQL databases. The results were presented in Table 4.2.

Table 4.2 NoSQL and Quality Attributes.

	Disagree		**Neither agree nor disagree**		**Agree**			
Quality Attribute	**N**	**%**	**N**	**%**	**N**	**%**	**Mean (M)**	**Standard Deviation (SD)**
Scalability	3	5.08%	5	8.47%	38	64.41%	4.33	0.99
Durability	3	5.08%	8	13.56%	35	59.32%	4.02	0.95
Efficiency	3	5.08%	9	15.25%	34	57.63%	4.02	0.98
Availability	4	6.78%	6	10.17%	36	61.02%	3.96	0.94
Consistency	4	6.78%	9	15.25%	33	55.93%	3.89	0.97
Maintainability	5	8.47%	9	15.25%	32	54.24%	3.87	1.02
Recovery Time	4	6.78%	8	13.56%	34	57.63%	3.85	0.99
Reliability	5	8.47%	12	20.34%	29	49.15%	3.80	1.05

From the Table 4.2 and Figure 4.2, it is observed that different results were obtained than the results for SQL comparison. 64.41% of the participants agreed that scalability is an important attribute that is affecting the NoSQL databases with a 4.33 mean. One of the powerful features of NoSQL databases is achieving high scalability to handle the high workloads [34]. NoSQL databases are horizontally scalable, hence by adding more servers to the database, the increased workload can be handled. NoSQL databases can become more extensive and powerful, making them the preferred choice for large and rapidly changing datasets.

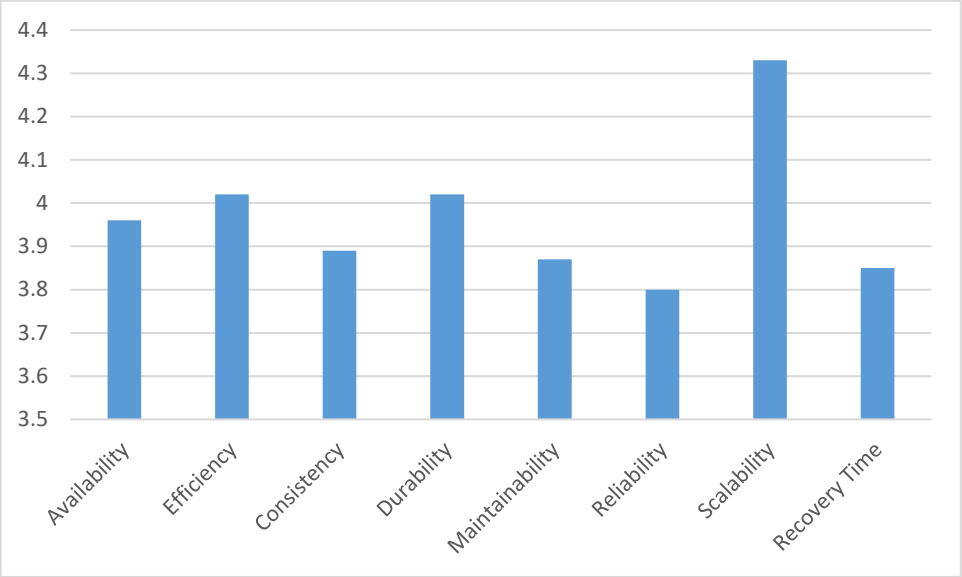

Figure 4. 2 NoSQL and Quality Attributes Mean Scores.

After that, efficiency and durability have the highest mean scores compared to the other attributes, with a 4.02 mean. Thirty-four respondents agreed that efficiency is an important quality attribute for NoSQL databases, whereas 35 agreed with durability. In the literature, efficiency is also reported as one of the top quality attributes that many NoSQL software built about because of its simplicity and efficiency in clustering data [32]. The questionnaire results show that durability is one of the main effective quality attributes, even though NoSQL is not based on the ACID properties. So, to improve durability, different methods should be used in NoSQL systems to ensure high durability, such as synchronously replicating writes across the three replicas or data logging mechanisms.

61.02% of respondents agreed that availability is effective in NoSQL databases, with a 3.96 mean score. NoSQL emphasizes "Availability" and "Performance" [35], while the focus of BASE in main terms is permanent availability, so it provides a high state of availability by allowing partial failure instead of a complete system failure. Data availability in NoSQL is weak when cloud-server approach is used in such systems [32].

After that, consisency comes in the fifth order, with 33 respondetns agreeing that this attibute has an effect on NoSQL databases with a 3.89 mean. According to the literature findings, SQL databases have stable, strong consistency, whereas NoSQL databases sacrifice consistency for availability. BASE offers availability while loosening the strict consistency requirements.

After consistency, we see maintainability with total of 32 agree votes and 3.87 mean. It is clear that not many participants believe that maintainability is a very effective quality aspect of NoSQL, most NoSQL systems offer limited maintainability when compared with traditional RDBMS [18]. According to many tests results, no NoSQL system was found to achieve optimal results.

For recovery time with a 3.85 mean, the resilient architecture of NoSQL databases typically provides a buffer against data loss; MongoDB and Aerospike are good picks if looking for good recovery time [2].

According to the results of the questionnaire, the least effective quality factor in the participants' perspective for NoSQL was reliability with a total of 29 agree votes and 3.8 mean. Some literature studies reported a lack of reliability in NoSQL databases because eventual consistency is provided by BASE properties, in which the time between actions of one transaction could vary in the execution time [20].

In the second part of the questionnaire, the participants were asked about which of the eight quality attributes they think are important for relational databases (SQL) to have an idea about the importance of each quality attribute, given in Table 4.3.

Table 4.3 Importance of Quality Attributes in SQL

Quality Attribute	**N**	**%**
Maintainability	29	49.15%
Efficiency	26	44.07%
Consistency	25	42.37%
Availability	21	35.59%
Durability	20	33.90%
Scalability	20	33.90%
Reliability	19	32.20%
Recovery Time	15	25.42%

As seen from Table 4.3, maintainability got the highest votes with a total of 29 votes (49.15%), then efficiency with 26 votes (44.07%), and consistency with 25 total votes (42.37%). For the least important quality attribute, they voted for recovery time with 15 votes (25.42%).

The same question was asked about which of the eight quality attributes they think are important for non-relational databases. The results were provided to have an idea about the importance of each quality attribute from their perspective, given in Table 4.4.

Table 4.4 Importance of Quality Attributes in NoSQL

Quality Attribute	N	%
Consistency	28	47.46%
Efficiency	27	45.76%
Reliability	27	45.76%
Availability	26	44.07%
Scalability	26	44.07%
Maintainability	25	42.37%
Recovery Time	22	37.29%
Durability	19	32.20%

From what we can notice in Table 4.4, the results are different from what we got for the same question about relational databases. For non-relational databases, the most important quality attribute from the perspective of the participants was consistency with a total of 28 votes (47.46%), then both efficiency and reliability got the same amount of votes with total of 27 votes each (45.76%), and then availability and scalability both got 26 total votes each (44.07%). While for the least important quality attribute, they voted for durability with a total of 19 votes (32.30%).

Additionally, an open-ended question was asked to understand the IT professionals' perceptions and opinions regarding their preferences for the databases. The answers to the question " What would you like to add / suggestions about relational databases and non-relational database preferences in your institution?" are discussed below. This

question is optional, so a total of 13 responses were recorded for this question. The questionnaire participants are represented by their IDs, which are automatically assigned by the survey application (i.e. P1, P2, etc.) as an autoincrement number uniquely given to each participant.

Three of the respondents claim that the preference of the database depends on the project, and P22 reported that the selection should be made according to the application. Similarly, P37 stated, "In fact, I believe that these two types of [relational and non-relational] databases can be preferred according to the scope and purpose of the project. Although relational databases are at the forefront regarding recovery time, consistency, and maintainability, I think NoSQL is more flexible in terms of scalability and efficiency." P48 also mentioned that it is crucial to choose the proper method for our project according to the company's needs.

Four of the respondents reported that they preferred SQL mostly. According to P5, "I prefer using SQL because it's more widely used and easier to find and fix its problems". P4 stated that only SQL databases are used in the company they work in. P7 said it is better to use the SQL systems due to the high community support. Similarly, P12 added, "A relational database organizes data into tables which can be linked—or related—based on data common to each. This capability enables you to retrieve a new table from data in one or more tables with a single query. It also allows you and your business to understand the relationships among all available data better and gain new insights for making better decisions or identifying new opportunities".

On the other hand, four respondents reported that they preferred NoSQL mostly. P47 said, "In my experience, NoSQL has been very stable in performance, especially for large amounts of data," and that MySQL is a good program for medium and small businesses. Similarly, P6 and P8 thought that, according to their experience, NoSQL is better for larger companies and a high amount of data. P10 claimed that their company is currently using SQL, but in his opinion, NoSQL is better for medium and larger companies.

Two respondents (P9 and P21) claimed both relational and non-relational databases could be used together based on the need of the project. For the tools used in the IT

companies, P21 added that considering relational database software and tools, mostly PostgreSQL and Oracle are preferred. In contrast, for the non-relational databases, Redis and MongoDB are used in the current projects. P21 informed that “IT companies should also take into account the costs while making these choices. I think that the biggest factor in the widespread use of PostgreSQL lately is that it is free.”. He also suggested that in education, it is necessary to learn the use of the Jquery database and in-depth information about the use of MLSQL [A type of SQL developed for Big Data & AI applications]. Training in a laboratory environment during education is required for database migration, and practice should be done when merging the data from different databases.

CHAPTER 5

CONCLUSION

This study aimed to distinguish the effects of the main eight software quality attributes on the choice between SQL and NoSQL DBMS. By mentioning the quality attributes and observing the theoretical and practical points of view that either support or reject the DBMS from being used according to the highest quality attributes priorities for the work.

The study results showed that it is preferred to use SQL DBMS if your business considers the highest important quality attributes are durability, consistency, and availability. This is an expected result since SQL databases are based on ACID properties, so they have an overall good result of supporting those properties. According to the questionnaire results, the Maintainability and Recovery Time got low scores for the SQL databases. In addition, it can be inferred that if high scalability is needed for your work, SQL databases may not perform better.

Considering NoSQL databases, the highest important quality attributes are scalability, durability, and efficiency according to the questionnaire results. The literature also supports these results that NoSQL provides high scalability due to its horizontal scaling feature.

The main contribution of this thesis work is to provide insight into the importance of deciding the main essential quality attributes for the projects, then deciding the most suitable DBMS type accordingly based on the needs of the specific projects. The results reveal that it is crucial to make a correct choice to avoid high costs in the database quality, which may lead to future waste of resources and is also vital to reduce the cost.

5.1 Limitations of Study

The characteristics of the sample can be reported as a limitation of this study. The survey participants were mainly from a single department (computer engineering). So further analysis cannot be performed based on the group differences such as departments and experience levels. The participant distribution cannot be controlled since it is distributed online. In this study, the number of highly experienced professionals is very low. Another limitation can be the time which can be extended to reach more participants.

5.2 Future Work

This study is conducted to understand software quality attributes' effects on DBMS from mostly students' perspectives. In the future, it may be possible to enhance the quality of results by reaching out to highly experienced domain experts working in the field of databases. Interviews can be conducted with such experts to get more detailed information, enabling to compare the results of the questionnaire, the findings of the literature, and interview results for validity and reliability of the study. More emphasis can be given to the specific analysis of the different quality attributes, such as read and write performance, to differentiate between SQL and NoSQL by more factors.

REFERENCES

[1] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, and J. Bernardino, "Choosing the right NoSQL database for the job: a quality attribute evaluation," *J. Big Data*, vol. 2, no. 1, pp. 1–26, 2015.

[2] J. R. Lourenço, V. Abramova, M. Vieira, B. Cabral, and J. Bernardino, "Nosql databases: A software engineering perspective," in *New Contributions* in *Information Systems and Technologies*, 2015, pp. 741–750.

[3] E. F. Codd, A Relational Model of Data for Large Shared Data Banks. Berlin, Germany: Springer, 2002, pp. 263–294.

[4] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems 7 th Edition. Pearson Harlow, 2017, pp. 291.

[5] S. K. Giri, V. Chidanandan, and N. Rajagopalan, "Research of NOSQL and SQL and designing a better database schema for databases," Int. J. Eng. Adv. Technol., vol. 8, no. 6 Special Issue, pp. 177–182, 2019.

[6] S. Venkatraman, K. Fahd, S. Kaspi, and R. Venkatraman, "SQL versus NoSQL movement with big data analytics," Int. J. Inf. Technol. Comput. Sci., vol. 8, no. 12, pp. 59–66, 2016.

[7] N. Q. Mehmood, R. Culmone, and L. Mostarda, "Modeling temporal aspects of sensor data for MongoDB NoSQL database," J. Big Data, vol. 4, no. 1, pp. 1–35, 2017.

[8] G. Antoniou, "Evaluation of Major NoSQL Databases." MSc Dissertation, The University of Edinburgh, Scotland, 2016.

[9] A. Nayak, A. Poriya, and D. Poojary, "Type of NOSQL databases and its

comparison with relational databases," Int. J. Appl. Inf. Syst., vol. 5, no. 4, pp. 16–19, 2013.

[10] E. Bertino and R. Sandhu, "Database security-concepts, approaches, and challenges," IEEE Trans. Dependable Secur. Comput., vol. 2, no. 1, pp. 2–19, 2005.

[11] X. Xu, Y. Lei, and Z. Li, "An incorrect data detection method for big data cleaning of machinery condition monitoring," IEEE Trans. Ind. Electron., vol. 67, no. 3, pp. 2326–2336, 2019.

[12] W. Ali, M.U. Shafique, M.A. Majeed, and A. Raza, "Comparison between SQL and NoSQL databases and their relationship with big data analytics," Asian J. Res. Comput. Sci., vol. 4, no. 2, pp.1-10, 2019.

[13] R. P. Padhy, M. R. Patra, and S. C. Satapathy, "RDBMS to NoSQL: reviewing some next-generation non-relational database's," Int. J. Adv. Eng. Sci. Technol., vol. 11, no. 1, pp. 15–30, 2011.

[14] P. D. Kaur, A. Kaur, and S. Kaur, "Performance analysis in bigdata," Int. J. Inf. Technol. Comput. Sci., vol. 7, no. 11, pp. 55–61, 2015.

[15] A. Prasad and B. N. Gohil, "A Comparative Study of NoSQL Databases," Int. J. Adv. Res. Comput. Sci., vol. 5, no. 5, pp. 171–174, 2014.

[16] Y.-L. Choi, W.-S. Jeon, and S.-H. Yoon, "Improving database system performance by applying NoSQL," J. Inf. Process. Syst., vol. 10, no. 3, pp. 355–364, 2014.

[17] M. Riaz, E. Mendes, and E. Tempero, "Towards predicting maintainability for relational database-driven software applications: Extended evidence from software practitioners," Int. J. Softw. Eng. Its Appl., vol. 5, no. 2, pp. 107–122, 2011.

[18] S. S. Pore and S. B. Pawar, "Comparative study of SQL & NoSQL databases," Int. J. Adv. Res. Comput. Eng. Technol., vol. 4, no. 5, pp. 1747–1753, 2015.

[19] M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, “Relational vs. nosql databases: A survey,” Int. J. Comput. Inf. Technol., vol. 3, no. 03, pp. 598–601, 2014.

[20] A. H. Al Hinai, “A Performance Comparison of SQL and NoSQL Databases for Large Scale Analysis of Persistent Logs.” M.A. thesis, Uppsala University, Sweden, 2016.

[21] Solid IT. gmbh, “DB-Engines Ranking," Internet: https://db-engines.com/en/ranking, [29 May 2022].

[22] J. Phipps, “IBM Db2,” Internet: https://www.webopedia.com/definitions/ibm-db2/, 18 February 2021 [29 May 2022].

[23] “What is Microsoft Access? - History & Overview,” Internet: https://study.com/academy/lesson/what-is-microsoft-access-history-overview.html, 11 April 2022 [30 May 2022].

[24] X. Yang, “Analysis of DBMS: MySQL Vs PostgreSQL.” The Bachelor’s Thesis, Kemi-Tornio University, Finland, 2011

[25] “What is PostgreSQL? Robust, reliable open source relational database,” Internet: https://aws.amazon.com/rds/postgresql/what-is-postgresql/, [30 May 2022].

[26] A. Makris, K. Tserpes, D. Anagnostopoulos, M. Nikolaidou, and J. A. F. de Macedo, “Database system comparison based on spatiotemporal functionality,” in IDEAS 2019: 23rd International Database Engineering & Applications Symposium, 2019, pp. 1–7.

[27] C. Băzăr and C. S. Iosif, “The transition from rdbms to nosql. a comparative analysis of three popular non-relational solutions: Cassandra, mongodb and couchbase,” Database Syst. J., vol. 5, no. 2, pp. 49–59, 2014.

[28] A. Makris, K. Tserpes, V. Andronikou, and D. Anagnostopoulos, “A classification of NoSQL data stores based on key design characteristics,”

Procedia Comput. Sci., vol. 97, pp. 94–103, 2016.

[29] “What Is Redis and Why Is It so Popular?,” Internet: https://blog.eduonix.com/web-programming-tutorials/what-is-redis-and-why-is-it-so-popular/, 22 July 2019 [29 May 2022].

[30] “Redis vs. MongoDB: What you need to know,” Internet: https://www.fosslinux.com/49916/redis-vs-mongodb.htm, 19 January 2022 [29 May 2022].

[31] V. Sharma and M. Dave, “Sql and nosql databases,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 2, no. 8 , pp. 20–22, 2012.

[32] D. Hammes, H. Medero, and H. Mitchell, “Comparison of NoSQL and SQL Databases in the Cloud,” Proc. South. Assoc. Inf. Syst. (SAIS), Macon, GA, pp. 21–22, 2014.

[33] “SQL to NoSQL: Architecture Differences and Considerations,” SCYLLADB, vol. 1, pp. 6–8, 2020.

[34] M. Rys, “Scalable SQL: How do large-scale sites and applications remain SQL-based?,” Queue, vol. 9, no. 4, pp. 30–37, 2011.

[35] D. G. Chandra, “BASE analysis of NoSQL database,” Futur. Gener. Comput. Syst., vol. 52, pp. 13–21, 2015.

APPENDIX A

QUESTIONNAIRE IN TURKISH

SQL ve NoSQL Veritabanı Tercihlerine Yönelik Anket Çalışması

Bu anket çalışması “Evaluating the Quality Aspects of SQL and NoSQL Databases / SQL ve NoSQL Veritabanlarının Kalite Açısından Değerlendirilmesi” başlıklı çalışma kapsamında, SQL ve NoSQL veritabanları ile ilgili Bilgisayar / Bilişim Sistemleri / Yazılım Mühendisliği öğrencilerinin ve sektördeki alan uzmanlarının kalite kapsamında deneyimlerinin anlaşılmasına yönelik olarak geliştirilmiştir.

Bu anket Atılım Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümünde yürütülmekte olan bir yüksek lisans tez çalışması kapsamında SQL ve NoSQL veritabanlarının kalite ölçütleri açısından öğrencilerin ve sektördeki alan uzmanlarının deneyimlerinin anlaşılmasına yönelik olarak hazırlanmıştır. Ankette katılımcılardan kimlik belirleyici hiçbir bilgi istenmemektedir. Cevaplarınız tamamıyla gizli tutulacak ve sadece araştırmacılar tarafından değerlendirilecektir; elde edilecek bilgiler tez ve bilimsel yayımlarda kullanılacaktır.

Bu çalışmaya tamamen gönüllü olarak katılıyorum ve istediğim zaman yarıda kesip çıkabileceğimi biliyorum. Verdiğim bilgilerin bilimsel amaçlı yayımlarda kullanılmasını kabul ediyorum.

Değerli katkılarınız için şimdiden teşekkür ederiz.

Bölüm-1.Temel Bilgiler

1. Hangi yaş grubundasınız?

- 18 - 25
- 26 - 33
- 34 - 41

- 42 - 49
- 50 ve üzeri

2. Cinsiyetiniz nedir?

- Erkek
- Kadın

3. En son edindiğiniz diploma derecesi nedir?

- Önlisans
- Lisans
- Yüksek lisans
- Doktora

4. Mezun olduğunuz programı/programları seçiniz.

- Bilgisayar Mühendisliği
- Yazılım Mühendisliği
- Bilişim Sistemleri Mühendisliği
- Bilgisayar Bilimleri
- Önlisans
- Diğer:

5. Öğrenci iseniz şu an okumakta olduğunuz bölümü seçiniz.

- Bilgisayar Mühendisliği
- Yazılım Mühendisliği
- Bilişim Sistemleri Mühendisliği
- Bilgisayar Bilimleri
- Önlisans
- Diğer:

6. Çalıştığınız şirketteki pozisyonunuz nedir?

- Geliştirici (Developer)
- Proje yöneticisi (Project Manager)

- Takım lideri (Team Leader)
- Test mühendisi (Test Engineer)
- Sistem çözümleyici (System Analyst)
- Sistem mimarı (System Architect)
- Sistem tasarımcısı (System Designer)
- Diğer:

7. Proje yöneticisi veya takım lideri olarak kaç yılınızı harcadınız?

- Proje yöneticisi / takım lideri olarak çalışmadım
- 5 yıldan az
- 6 - 10 yıl
- 11 - 15 yıl
- 15 yıldan fazla

8. Mesleğinizi kaç yıldır icra ediyorsunuz? *

- 0 - 3 yıl
- - 6 yıl
- 7 - 10 yıl
- 10 yıldan fazla

Bölüm-2.1 SQL Veritabanı Kalite Öznitelikleri

1. Aşağıdaki tabloda verilen kalite özniteliklerinin ilişkisel veritabanları (SQL) açısından ne derece etkin olduğunu değerlendirip işaretleyiniz (1: En Düşük, 5: En Yüksek).

Kalite Özniteliği:	**1**	**2**	**3**	**4**	**5**
Kullanılabilirlik (Availability)					
Verimlilik (Efficiency)					
Tutarlılık (Consistency)					
Süreklilik (Durability)					
Bakımı yapılabilirlik (Maintainability)					
Güvenilirlik (Reliability)					

Ölçeklenebilirlik (Scalability)					
Kurtarma Süresi (Recovery Time)					

Bölüm-2.2 NoSQL Veritabanı Kalite Öznitelikleri

1. Aşağıdaki tabloda verilen kalite özniteliklerinin ilişkisel olmayan veritabanları (NoSQL) açısından ne derece etkin olduğunu değerlendirip işaretleyiniz (1: En Düşük, 5: En Yüksek).

Kalite Özniteliği:	**1**	**2**	**3**	**4**	**5**
Kullanılabilirlik (Availability)					
Verimlilik (Efficiency)					
Tutarlılık (Consistency)					
Süreklilik (Durability)					
Bakımı yapılabilirlik (Maintainability)					
Güvenilirlik (Reliability)					
Ölçeklenebilirlik (Scalability)					
Kurtarma Süresi (Recovery Time)					

Bölüm-2.3 SQL için Kalite Özniteliklerinin Önemi

1. Aşağıda verilen kalite özniteliklerinden hangilerinin ilişkisel veritabanlarının geliştirilmesi için (SQL) önemli olduğunu düşünüyorsunuz? (birden çok seçenek işaretlenebilir)

- Kullanılabilirlik (Availability)
- Verimlilik (Efficiency)
- Tutarlılık (Consistency)
- Süreklilik (Durability)
- Bakımı yapılabilirlik (Maintainability)
- Güvenilirlik (Reliability)
- Ölçeklenebilirlik (Scalability)
- Kurtarma Süresi (Recovery Time)

Bölüm-2.4 NoSQL için Kalite Özniteliklerinin Önemi

1. Aşağıda verilen kalite özniteliklerinden hangilerinin ilişkisel olmayan veritabanlarının geliştirilmesi için (NoSQL) önemli olduğunu düşünüyorsunuz? (birden çok seçenek işaretlenebilir)

- Kullanılabilirlik (Availability)
- Verimlilik (Efficiency)
- Tutarlılık (Consistency)
- Süreklilik (Durability)
- Bakımı yapılabilirlik (Maintainability)
- Güvenilirlik (Reliability)
- Ölçeklenebilirlik (Scalability)
- Kurtarma Süresi (Recovery Time)

2. Çalıştığınız kurumdaki tecrübeleriniz ışığında ilişkisel veritabanları ve ilişkisel-olmayan veritabanı tercihleri ile ilgili varsa eklemek istedikleriniz / önerileriniz nelerdir?

Lütfen yanıtınızı buraya yazın.

APPENDIX B

QUESTIONNAIRE IN ENGLISH

A Survey on SQL and NoSQL Database Preferences

This survey is conducted based on a study titled as “Evaluating the Quality Aspects of SQL and NoSQL Databases / SQL ve NoSQL Veritabanlarının Kalite Açısından Değerlendirilmesi”, to understand the experiences and perceptions of Computer / Information Systems / Software Engineering students and domain experts in IT sector on the quality aspects of SQL and NoSQL databases.

This study was carried out by the member of Atılım University, Faculty of Engineering, Department of Computer Engineering, Dr. Damla Topalli, to understand the students’ and domain experts’ experiences and perceptions on the quality aspects of SQL and NoSQL databases titled as ““Evaluating the Quality Aspects of SQL and NoSQL Databases / SQL ve NoSQL Veritabanlarının Kalite Açısından Değerlendirilmesi”. An online questionnaire regarding to this study is implemented. If you agree to participate in and complete the study, all personal information, responses to the questionnaire will be treated confidentially and to be evaluated only by the researchers; the findings will be reported in thesis and publications.

I understand that my participation is voluntary and that I may withdraw from the study at any time without explanation. If you agree to take part in this study, please click on the “Next” button and start survey.

Thank you in advance for your voluntary participation in this study.

Part-1.General Information

1. Select your age group:

- 18 - 25
- 26 - 33
- 34 - 41
- 42 - 49
- 50 and above

2. Select your gender:

- Male
- Female

3. What is your latest diploma degree?

- Associate degree
- Bachelors (BSc)
- Masters (MSc)
- Doctor of Philosophy (PhD)

4. Select program(s) that you have graduated from:

- Computer Engineering
- Software Engineering
- Information Systems Engineering
- Computer Sciences
- Associate degree
- Other:

5. If you are a student, select the deparment you are currently studing:

- Computer Engineering
- Software Engineering
- Information Systems Engineering
- Computer Sciences
- Associate degree
- Other:

6. If you are working in a company, please select your position:

- Developer
- Project Manager
- Team Leader
- Test Engineer
- System Analyst
- System Architect
- System Designer
- Other:

7. How many years have you spent as a project manager or team leader?

- I did not work as a project manager / team leader
- Less than 5 years
- 6 - 10 years
- 11 - 15 years
- More than 15 years

8. How many years have you been practicing your profession?

- 0 - 3 years
- 4 - 6 years
- 7 - 10 years
- More than 10 years

Part-2.1 SQL Database Quality Attributes

1. Evaluate and select the option that suits you best in terms of SQL databases to indicate how effective these quality features are in SQL databases (1: Lowest, 5: Highest).

Quality Attribute:	**1**	**2**	**3**	**4**	**5**
Availability					
Efficiency					
Consistency					
Durability					

Maintainability					
Reliability					
Scalability					
Recovery Time					

Part-2.2 NoSQL Database Quality Attributes

1. Evaluate and select the option that suits you best in terms of NoSQL databases to indicate how effective these quality features are in NoSQL databases (1: Lowest, 5: Highest).

Quality Attribute:	**1**	**2**	**3**	**4**	**5**
Availability					
Efficiency					
Consistency					
Durability					
Maintainability					
Reliability					
Scalability					
Recovery Time					

Part-2.3 Importance of Quality Attributes for SQL

1. Which of the following quality attributes do you think are important for relational databases (SQL)? (multiple options can be selected)

- Availability
- Efficiency
- Consistency
- Durability
- Maintainability
- Reliability
- Scalability

- Recovery Time

Part-2.4 Importance of Quality Attributes for NoSQL

1. Which of the following quality attributes do you think are important for non-relational databases (NoSQL)? (multiple options can be selected)

- Availability
- Efficiency
- Consistency
- Durability
- Maintainability
- Reliability
- Scalability
- Recovery Time

2. What would you like to add / suggestions about relational databases and non-relational database preferences in your institution?

Please write your answer here.