12-2021

Meta-analysis Of Performance Characteristics of Modern Database Schemas

Carter Grove

META-ANALYSIS OF PERFORMANCE CHARACTERISTICS OF MODERN DATABASE SCHEMAS

THESIS

Carter E. Grove, GG-12, DAF

AFIT-ENV-MS-D-047

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENV-MS-D-047

META-ANALYSIS OF PERFORMANCE CHARACTERISTICS OF MODERN DATABASE SCHEMAS

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Carter E. Grove, BS

GG-12, DAF

December 2021

AFIT-ENV-MS-D-047

META-ANALYSIS OF PERFORMANCE CHARACTERISTICS OF MODERN DATABASE SCHEMAS

Carter E. Grove, BS

GG-12, DAF

Committee Membership:

Brent T. Langhals, PhD
Chair

Paul M. Beach, PhD, CISSP
Member

Michael R. Grimaila, PhD, CISM, CISSP
Member

Abstract

Industry and academia alike use databases to solve advanced and complex problems. A large variety of database types exist, each with different advantages or disadvantages depending upon user needs. To understand which database schema is best suited for a given user's needs, this study explored how databases are measured against each other, what relevant performance characteristics exist, and what advantages each type of database inherently possesses. To accomplish this task, a meta-analysis of over 50 articles was conducted. The results of each study was aggregated to determine which database schemas exhibited the best performance for accuracy, scalability, transactions, query latency, and writing latency. The results indicate NoSQL databases performed the best for scalability, transactions, and query and writing latency, making them advantageous for database solutions for unique problems. Relational databases, however, provided the best accuracy among databases and were often the cheapest solution, making them suitable for basic database needs.

Acknowledgments

I would like to thank my faculty advisor, Dr. Brent Langhals for his support, guidance and flexibility throughout this thesis effort. His leadership and experience throughout this effort was unapparelled and highly appreciated. I also want to thank the Signals Analysis Squadron of the National Air and Space Intelligence for supporting my academic endeavors.

Carter E. Grove

Table of Contents

Page

List of Figures

Page

List of Tables

Page

META-ANALYSIS OF PERFORMANCE CHARACTERISTICS OF MODERN DATABASE SCHEMAS

I. Introduction

Background

The past few decades have witnessed a technological explosion that traditional data management practices have been struggled to keep up with. Perhaps the best example of this is database creation, management, and querying. Over the past 20 years over 200 new types of databases have been created, each having different characteristics and attributes that make them more or less ideal for certain solutions depending on user needs (Fan, 2016). Additionally, with the emerging prevalence of big data in virtually all industries, motivation to optimize the usage of different database types has increased due to the sheer volume of data that is utilized in the modern world. (Hossain, 2013)

Traditionally, relational databases were preferred as they could employ ACID (atomicity, consistency, isolation, durability) principles and guaranteed data validity. Unfortunately, this type of database is ill-suited for big data solutions. For example, a Google search using BigTable (a type of NoSQL Wide-Column database), is capable of scaling to billions of rows and thousands of columns, enabling storage of terabytes or even petabytes of data (Google, 2021). A relational database simple does not have the capacity to match that capability with modern computing power. New NoSQL databases such as Graph, Document, Key-Value, and Wide-Column are all alternative database types that relax the ACID constraints and thus are better suited for big data solutions as well as supporting multiple users simultaneously. The primary research question is:

which database type (between SQL and multiple NoSQL options) is best for a given set of user needs? To study which of these data base types provide the best performance characteristics in different scenarios, a meta-analysis of scholarly work (peer-reviewed journal articles from the last ten years will be conducted.

As the differences between SQL and NoSQL are explored, it is expected that NoSQL will outperform SQL in most modern and robust applications, whereas SQL will likely remain the database schema of choice for more traditional databases without extenuating requirements. However, "NoSQL" only indicates the absence of a relational database, so as NoSQL takes over the modern database landscape, it is not clear which NoSQL database schema is most advantageous for certain applications. This research is attempting to generate and understand patterns that will help determine how to choose which NoSQL database schema is best for any given application. Even though SQL likely will not have the performance characteristics necessary to remain competitive against NoSQL in a complex and modern environment, it will continue to be included in order to provide a baseline measuring point for all other databases.

Problem Statement

The problem facing both academia and industry today in database creation is the selection of database schemas to achieve the most effective performance for their desired needs, a common pitfall of which is selecting a database schema that does not appropriately scale or allow for implementation of changes in the data leading to costly overhauls at a later date. In order to mitigate these concerns, is it possible to develop

techniques that allow testing of different database schemas in an effort to fully understand the best database selection prior to creation.

Research Objectives/Questions/Hypotheses

An initial search of existing literature indicated the selection of the database type is highly dependent on the goals of the database use case (Moniruzzaman, 2013; Gupta, 2017). For example, relational databases (SQL) while powerful for storing structured data and capable of executing complex queries, experiences serious performance issues when the use case requires extremely large amounts of heterogenous unstructured data. When large data sets are introduced and partitioning is required, the time and processing power required to operate relational databases increases at a faster rate than that of NoSQL databases, making NoSQL a better option for any database that will fall into the category of "Big Data" (Sánchez-de-Madariaga, 2017; Wang, 2019). However, each NoSQL class of databases is typically optimized to meet other user priorities such as reduced read or write latencies.

The challenge facing both academia and industry today is the lack of clear research into the relative advantages/disadvantages of NoSQL database types, Therefore, the focus of this research will be to explore the body of literature since 2010 to determine if certain database types yield clear advantages by reviewing past database performance tests and comparisons. Specifically, this research will attempt to answer the questions listed below:

- How are databases measured against each other?

- Can the performance characteristics of the different database schemas be meaningfully compared to each other?
- What advantages of the selected database schemas (Relational, Graph, Document, Key-Value, and Wide-Column) can be determined for different applications?

Methodology

This paper uses a meta-analysis of past literature across all the different types of database schemas. Fifty articles were selected and aggregated to form the basis of data to analyze. The data included results from tests on both real data, and simulated data. The data was then used to compare which database schemas exhibited the best performance for Accuracy, Scalability, Transactions (or Volume), Query Latency and Writing Latency.

Assumptions/Limitations

The scope of this research is limited five types of performance characteristics Accuracy, Scalability, Transactions or Volume, Query Latency and Writing Latency. While other characteristics exist and may be important for specific use cases, the five selected are the most commonly examined performance characteristics today. Additionally, due to swift changes in technologies, this research's relevance may be limited to a relatively short period of time.

Since this research was a meta-analysis, the data used was second hand data. Therefore, we are unable to independently verify the data was collected properly and

must assume proper steps were taken by past researchers to ensure the data was not altered by poor collection processes.

Conclusion

Our goal is to determine the advantages and disadvantages in database performance characteristics among the five database schemas. To do this we will be conducted a meta-analysis of past research. Later in this paper, we will discuss what literature was used for the analysis and why, how the analysis was conducted, and the results of the analysis. Lastly, we will discuss what these results mean for future researchers in the database field as well as those who intend to construct new or implement existing database technologies.

II. Literature Review

Databases

Databases can be organized and created in different ways that dictate how data is inserted, stored, and retrieved. These different organizational designs are referred to as schemas (Kolonko, 2018). At the broadest level, there are two types of databases schemas, Relational databases that commonly employ a Structured Query Language (SQL) as their interface and may also be referred to as simple SQL databases (as opposed to non-relational databases which are typically referred to as NoSQL databases). Relational databases are the traditional schema and are categorized by a set of tables where data gets fit into a pre-defined category. The table consists of rows and columns where the column has an entry for data for a specific category and rows contains instances for that data defined according to the category (Gupta, 2017). NoSQL databases on the other hand, do not follow this fixed and pre-defined mold, and therefore can be dynamic, support unstructured data and have a greater ability to adapt to changes (Abramova, 2014). To further break down the schema database types, there are four principal schemas for NoSQL databases, Key-Value, Document, Graph, and Wide-Column each of which will be further discussed in detail.

Relational Databases

As previously mentioned, a Relational Database is a database which follows a more traditional schema where data is held in predefined tables with rows and columns. Within this table, each column will hold a specific attribute of the data. For example, in

an inventory database at a car dealership, one column would hold the model of the car, another column would hold the year, etc. There will also be a column that holds a unique identifier for each row, known as a key. In this same example, this might be a one up counter of the vehicles as they are placed in inventory, or perhaps the VIN number. The data can then be queried based upon the key to yield all the data within the given row, or entire columns can be accessed showing all the different data based upon the selected attribute.

Advantages of Relational Databases include standardization with SQL and their ability to employ ACID principles to ensure data accuracy. However, their drawbacks include costly hardware required to operate if the size of the database is vastly increased, and the effort to normalize (format the data to fit the required bounds of the database) existing data (Hammes, 2014).

Key-Value Databases

Key-Value databases are a type of database that uses a simple key-value method to store data as a collection of key-value pairs in which a key serves as a unique identifier. Both keys and values can be anything, ranging from simple objects to complex compound objects (Ali, 2019). Since the Key-Value schema does not require each input to fill a predetermined set of rows, it can optimize the amount of data stored better than a Relational database. This also offers Key-Value databases the ability to scale easier and flex to meet changing needs. However, with the lack of a defined structure, there are some drawbacks, Key-Value databases are unable to efficiently employ ACID principles.

Document Databases

A Document database is a type of database that is designed to store and query data using tags or other methods to relate the data to different values. This is similar to the Key-Value approach, but differs in that is uses the metadata of the stored documents as the identification rather than strings within the data itself (Henricsson, 2011). Additionally, Document databases allow for versatility in querying with the use of Application Programming Interfaces (API) which is software that helps link between the computer and the database. An API is similar to the more commonly known Graphical User Interface (GUI) except that the link is between computer and database rather than a human and a computer. Through the use of APIs, it is possible query within stored documents' content in addition to querying their metadata. While it is dependent upon how the database is setup to determine which document types it can store, the two most commonly used are JavaScript Object Notation (JSON) documents or Extensible Markup Language (XML) documents.

Graph Databases

A Graph database is a type of non-relational database that uses graph theory to store, map and query relationships. The relationships give the database the ability to link stored data together so it can be retrieved with a single operation. This can be complex for computing, but provides a more intuitive interface with the human user (Moniruzzaman, 2013).

Graph theory is based on relating pairs of data, referred to as nodes or vertices, using links referred to as edges. The definition of what constitutes these edges are dependent upon the database itself and can vary vastly among databases. For example, one Graph database could relate data using a mathematical algorithm to determine which nodes are connected by edges, and another could be based upon how many times the same user clicks on two different nodes.

Wide-Column Databases

A Wide-Column database is a database that stores data tables by column rather than by row. Essentially transposing the data in a Relational database. This allows the database to scan through specific columns of relevance within a dataset rather than scanning the entire dataset, and can easily discard unnecessary data (Dwivedi, 2012). Additionally, when all the data is aligned by columns, an entire set of rows can be assigned a single key allowing data compression, significantly reducing storage needs. For these reasons, Wide-Column databases make excellent candidates for vast scaling.

Database Characteristics

It is problematic to decisively determine that one database schema is a better performer over another because there are many different characteristics of a database that can be measured independently. For an example that is easy to visualize, you can have two different houses that are both the exact same square footage, but one is a ranch and one is three stories. The three-story house will be taller, the ranch will cover more

ground. They are both the same size; they just have advantages among different characteristics. The same principal applies to databases and the different characteristics each schema has.

There can be an overwhelming number of different characteristics to choose to measure. That is why it is necessary to focus on a few of the most prominent just like in the house example, square footage, bedrooms and bathrooms are the most common characteristics observed. For the purpose of this study, the characteristics in focus were narrowed down to the five most commonly observed among the literature and correlated to the most commonly sought-after performance parameters. The five characteristics are Accuracy, Scalability, Transactions or Volume, Query Latency, and Writing Latency, each of which we will discuss in further detail.

Accuracy

Accuracy refers to how likely the data stored within a database is correct. It is possible for to be considered inaccurate in a few ways. The first way, is the data can simple be missing where it should exist such as in accidental deletion or corruption. In this case, it will not be able to be retrieved in any manner. The next example of inaccurate data, is if it is inconsistent. The data in the example may be correct data, but is stored in a manner different from other similar types of data making it difficult to retrieve effectively when queried. To stick with the housing example, the square footage to one room may be entered as 100 square feet or 9.3 square meters. Both can be true, but the inconsistency between the way they are written can cause issues when attempting to access or view the

data. Lastly, data can be inaccurate if a database receives multiple entries for the same data store simultaneously. This can cause the data to exist, but the correct data could have been overwritten by slightly outdated data. In order to combat data inaccuracies, Relational databases employ the ACID principals to ensure absolute accuracy, however NoSQL databases must sacrifice some degree of accuracy in lieu of other characteristics.

Scalability

Scalability refers to the databases ability (or inability) to vastly increase in size and still perform adequately. This can mean in terms of the ability to hold the data itself without significant increases in latency time, and the ability of the database to manage more transactions simultaneously. Hardware clearly plays an important role on an individual databases ability to increase its data size or execute more transactions. However, for the purposes of studying databases, we are measuring the finite differences in an capability increase among the different database schemas given an equal hardware increase among all the schemas.

Transactions

A Transaction in terms of a database management system is one unit of work, or operation, for the database. This can be writing in a new data point, querying a datapoint, or editing a datapoint. We refer to the databases ability to conduct multiple of these units of works simultaneously as Transactions, or sometimes it is simply referred to as Volume of the data. In the context of this paper, Transactions encompasses two layers; the number

of operations a database can perform in a given time period, and the databases ability to execute those operations from multiple nodes or users simultaneously.

Query Latency

Query Latency is a straight forward measure of the time it takes for a database to execute a query. This is another characteristic that is seemingly tied directly to the hardware executing the query, and the size of the database. However, as mentioned before, different database schemas use very different methods to execute queries. Therefore, the specific time a query takes is only relevant for comparison to the same query using the same data and hardware, but from a different database schema.

Writing Latency

Similarly to the Query Latency, Writing Latency is a simple measure of the time it takes to write or insert data into the database. Again, the data size and hardware used are major factors, but the comparison between the different database schemas keeping all else equal is what is of interest.

Comparing Database Schemas

The database schemas described above will be compared against each other using an aggregation of data collected from previous research. We will use this data to try to determine how databases can be compared against each other, what performance characteristics are measurable and meaningful, and which database schemas will provide

the best performance for the specific application. Due to implementation of ACID principles employed by Relational databases, we expect Relational databases to exhibit the best accuracy. However, due to the characteristics of these same principles, we expect Relational database to perform the worse in the other four performance categories. The remaining four characteristics of query latency, writing latency, scalability, and transactions will likely be dominated by the NoSQL schemas. Sorting out which of the NoSQL has the best performance may prove more difficult due to the variability, but our initial presumption is that Key-Value database schemas will perform the best in the remaining categories due to its simplicity.

Summary

In this chapter we described the database schemas and how they function, leading to some of their assumed advantages and disadvantages. We also described the performance characteristics of databases and their importance. In the following chapters we will attempt to define a method for breaking down the database schemas and among the performance characteristics so that their advantages and disadvantages can be quantified and compared against each.

III. Methodology

Chapter Overview

The basis for this research is to perform an extensive literature review to compare the results of variety of previous studies to discern how different database schemas perform given varying types of data, volumes, or queries. The goal is to understand which database schemas would be best suited for a specific use case. While there are many different database sub-types, and the list is continually growing, we focused on the five classes of database schemas: Relational, Graph, Document, Key-Value, and Wide-Column.

Meta-Analysis

The research conducted in this paper is a meta-analysis. A meta-analysis is a statical examination of the results of many individual studies. The main objectives of a meta-analysis are to summarize and integrate results from a number of studies, analyze differences in results among the studies, increase sample sizes, determine if new studies are needed in a specific field, and generate hypothesis for future studies (Walker, Hernandez, & Kattan, 2008).

It is common for meta-analyses to comb over research that are geared towards answering the same question and analyzing the differences and similarities. However, in this case, there was not an abundance of published data geared directly towards the desired research questions. Therefore, the research in the relevant field was gathered, and

specific information pertaining to our research questions were teased out with the available data.

Article Selection and Validation

To begin the study, 50 journal articles were selected to form the basis of comparison. While it was not possible to "randomly" chose articles, a good faith effort was given to search for articles based only on relevance to the subject area and not include any biases. Nevertheless, it is possible some for some degree of biasness to be present due to the author;s accesses to (and thus emphasis) on peer-reviewed articles databases from the Institute of Electrical and Electronics Engineering (IEEE) and Association for Computing Machinery (ACM).

The articles selected compared different aspects of the database types including performance in both simulated and operational use. To understand the sample of the different database types, each article was reviewed and it was annotated which database schemas were studied in the article. These annotations were tabulated in spreadsheets to be used for comparison and analysis. Table 1 is a snapshot of the article titles along with which database schemas were evaluated.

Table 1: Database Articles with Schema Types

Title	Relational	Document	Key-Value	Graph	Wide-Column
A benchmark study on time series clustering					
A comparative study of elasticsearch and CouchDB document oriented databases		1			
A Comparative Study of Relational Database and Key-Value Database for Big Data Applications	1		1		
A Comparison of the Relative Performance of XML and SQL Databases in the Context of the Grid-SAFE Project	1	1			
A performance comparison of SQL and NoSQL databases	1	1	1		1
A Qualitative Comparison of NoSQL Data Stores		1	1	1	
A Quantitative Performance Analysis between Mongodb and Oracle NoSQL		1	1		
A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment	1	1			
An empirical comparison of graph databases				1	
Assessment of Graph Databases as a Viable Materiel Solution for the Army's Dynamic Force Structure (DFS) Portal Implementation		1		1	
BigQ: a NoSQL based framework to handle genomic variants in i2b2		1			
Capacity Measurement and Planning for NoSQL Databases			1		
Choosing the right NoSQL database for the job: a quality attribute evaluation		1	1		1
Comparative study of NoSQL databases for big data storage		1	1	1	1
Comparing NoSQL MongoDB to an SQL DB	1	1			
Comparing the Performance of NoSQL Approaches for Managing Archetype-Based Electronic Health Record Data		1			
Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics	1	1			
Comparison of NoSQL and SQL Databases in the Cloud	1	1			
Comparison of NoSQL Datastores for Large Scale Data Stream Log Analytics		1	1		1
Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications	1	1			
Comparison of SQL, NoSQL and NewSQL databases for internet of things	1	1			
Database Schema Matching Using Machine Learning with Feature Selection	1				
Designing performance monitoring tool for NoSQL Cassandra distributed database					1
Difference Between SQL Vs MySQL Vs SQL Server	1				
Distributed Relational Database Performance in Cloud Computing: an Investigative Study	1				
Document Oriented NoSQL Databases - A comparison of performance in MongoDB and CouchDB using a Python interface		1		1	
Evaluating the performance of SQL and NoSQL databases in an IoT environment	1	1			
Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NoSQL approaches	1	1			
Experimental Evaluation of NoSQL Databases		1	1	1	1
Load balancing for hybrid NoSQL database management systems		1	1		1
MongoDB vs MySQL: A Comparative Study on Databases	1	1			
NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison					
NoSQL databases: Critical analysis and comparison	1	1	1	1	1
NoSQL databases: MongoDB vs cassandra		1			1
Performance Analysis of Column Oriented Database versus Row Oriented Database	1				
Performance Analysis of Queries in RDBMS vs NoSQL	1	1			
Performance Comparison between Five NoSQL Databases		1	1		1
Performance Comparison of Relational Database with Object Database (DB4o)	1				
Performance comparison of the most popular relational and non-relational database management systems	1	1			
Performance Evaluation of MySQL and MongoDB Databases	1	1			
Perormance Evaluation of NoSQL Databases		1			1
Perormance Evaluation of NoSQL Systems, Using YCSB in a resource Austere Enviroment		1	1	1	
Quantitative Analysis of Scalable NoSQL Databases		1			1
Query Performance Analysis of NoSQL and Big Data	1	1			1
Review of NoSQL databases and performance testing on HBase		1	1		1
Scalable SQL and NoSQL Data Stores	1	1	1		1
Solving Big Data Challenges for Enterprise Application Performance Management		1	1		1
SQL Versus NoSQL Movement with Big Data Analytics					
SQL, noSQL or newSQL – comparison and applicability for Smart Spaces		1	1		
The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes		1			
	Totals				
	Relational	Document	Key-Value	Graph	Wide-Column
	23	38	17	8	16

For better visualization Figure 1 below illustrates what types of databases were evaluated in the selected articles. It should also be noted, that since many articles studied

more than one type of database schema, the cumulative list of the schemas adds up to more than the total number of articles evaluated. Among the articles chosen, Document-based database schemas were the most studied. It is unclear exactly why Document databases were studied the most, but anecdotally, it is likely due to the popularity of Document databases, especially MongoDB.

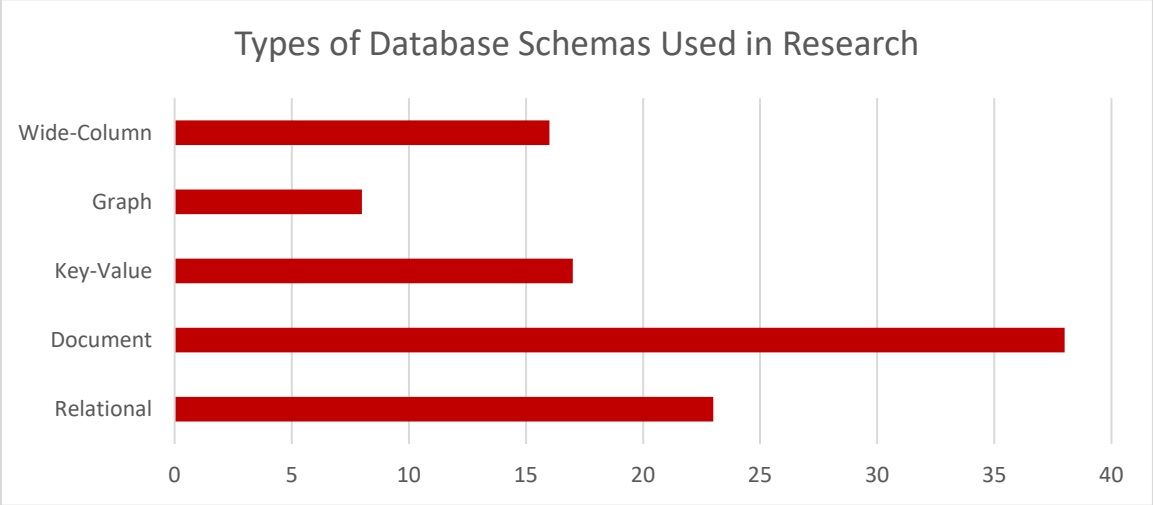

Figure 1: Database Articles by Type

Additionally, since there is a stark difference between Relational and NoSQL databases, it was important to document that of the fifty articles, 23 articles researched the differences between NoSQL databases and Relational databases, while 21 did not include any Relational databases.

After selecting the articles of interest and coding them based upon the database schemas, each article was further broken down into the parameters they studied and the type of method used to conduct the study. Due to the author's previous knowledge of databases, there were preconceived notions of what characteristics might be most commonly studied. However, it was still necessary to find which characteristics were

thought to be most valuable to the larger community. Therefore, each article was reviewed and notes were taken on each article annotating what aspects databases they studied or tested. These annotations were tabulated in spreadsheets to determine which characteristics were deemed relevant for analysis by previous researchers. The characteristics of interest were then narrowed down to the five most common: Querying Latency, Writing Latency, Transactions (Volume), Accuracy, and Scalability as defined in Chapter 2. Figure 3 below illustrates the breakdown of the articles among these five parameters and shows that query latency and writing latency were by far the two most common types of parameters studied.

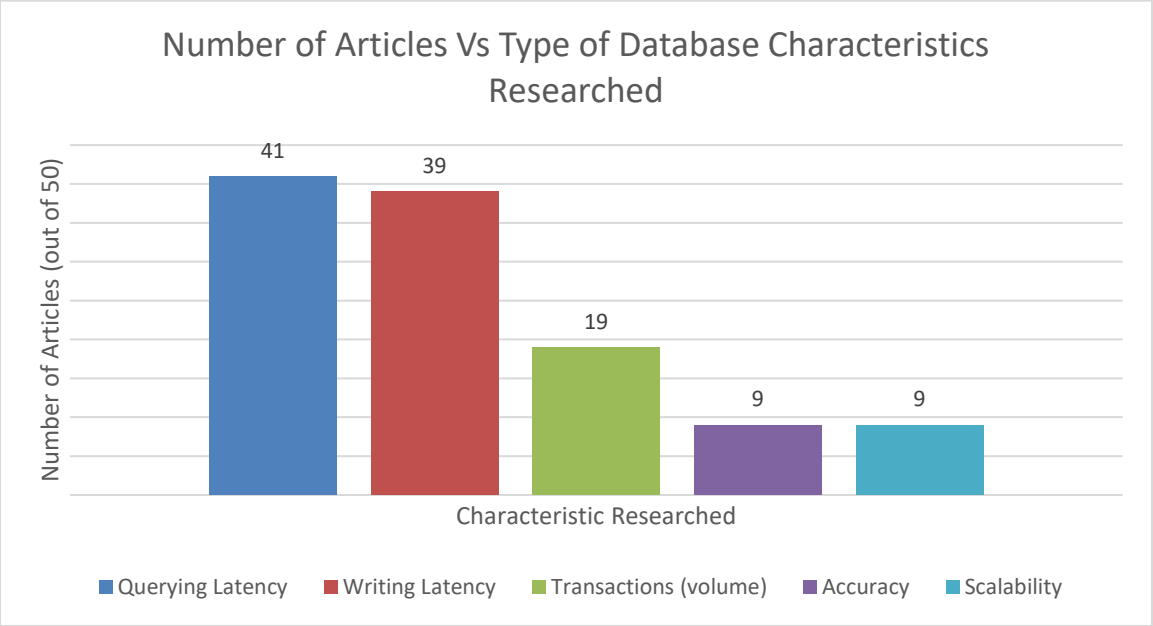

Figure 2: Database Parameters Compared

After determining what the most researched database schemas were and the most common performance characteristics studied, it became relevant to determine how the

research was conducted. The literature naturally broke into three primary categories coded as: Real, Simulations, Literature Review, with a few as Other.

- Real refers to testing on existing database schemas using real data pulled from either industrial or academic fields that are actively in use. This method typically was used to compare databases that are already operating for specific purposes. However, it often resulted in very limited studies due to the reduced range of tests that can be performed.
- Simulations refers to data that is artificially generated for the purposes of testing different database characteristics. This method provides better range of the types of tests that can be performed since the data was manipulated to suit the tests
- Literature Review refers to research that did not perform their own tangible testing on databases, but rather reviewed other research to develop conclusions.
- Other refers to the very few articles that did not fit into the other three categories and instead focused on developing tools for database testing, rather than the testing itself.

Once these categories were selected and defined, the articles were again reviewed and codified based upon which type of experiment or research they fell under. Table 2 shows a snapshot of a spreadsheet used to document this tabulation.

Table 2: Method of Database Comparison

	Type of Experiment			
Title	Real	Simulation	Literature Review	Other
A benchmark study on time series clustering				1
A comparative study of elasticsearch and CouchDB document oriented databases		1		
A Comparative Study of Relational Database and Key-Value Database for Big Data Applications		1		
A Comparison of the Relative Performance of XML and SQL Databases in the Context of the Grid-SAFE Project	1			
A performance comparison of SQL and NoSQL databases		1		
A Qualitative Comparison of NoSQL Data Stores	1			
A Quantitative Performance Analysis between Mongodb and Oracle NoSQL		1		
A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment		1		
An empirical comparison of graph databases		1		
Assessment of Graph Databases as a Viable Materiel Solution for the Army's Dynamic Force Structure (DFS) Portal Implementation		1		
BigQ: a NoSQL based framework to handle genomic variants in i2b2	1			
Capacity Measurement and Planning for NoSQL Databases		1		
Choosing the right NoSQL database for the job: a quality attribute evaluation			1	
Comparative study of NoSQL databases for big data storage			1	
Comparing NoSQL MongoDB to an SQL DB		1		
Comparing the Performance of NoSQL Approaches for Managing Archetype-Based Electronic Health Record Data	1			
Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics			1	
Comparison of NoSQL and SQL Databases in the Cloud	1			
Comparison of NoSQL Datastores for Large Scale Data Stream Log Analytics	1			
Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications		1		
Comparison of SQL, NoSQL and NewSQL databases for internet of things	1			
Database Schema Matching Using Machine Learning with Feature Selection				1
Designing performance monitoring tool for NoSQL Cassandra distributed database				1
Difference Between SQL Vs MySQL Vs SQL Server		1		
Distributed Relational Database Performance in Cloud Computing: an Investigative Study		1		
Document Oriented NoSQL Databases - A comparison of performance in MongoDB and CouchDB using a Python interface		1		
Evaluating the performance of SQL and NoSQL databases in an IoT environment		1		
Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NoSQL approaches	1			
Experimental Evaluation of NoSQL Databases		1		
Load balancing for hybrid NoSQL database management systems		1		
MongoDB vs MySQL: A Comparative Study on Databases		1		
NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison			1	
NoSQL databases: Critical analysis and comparison			1	
NoSQL databases: MongoDB vs cassandra		1		
Performance Analysis of Column Oriented Database versus Row Oriented Database		1		
Performance Analysis of Queries in RDBMS vs NoSQL		1		
Performance Comparison between Five NoSQL Databases		1		
Performance Comparison of Relational Database with Object Database (DB4o)	1			
Performance comparison of the most popular relational and non-relational database management systems	1	1		
Performance Evaluation of MySQL and MongoDB Databases		1		
Perormance Evaluation of NoSQL Databases		1		
Perormance Evaluation of NoSQL Systems, Using YCSB in a resource Austere Enviroment		1		
Quantitative Analysis of Scalable NoSQL Databases		1		
Query Performance Analysis of NoSQL and Big Data		1		
Review of NoSQL databases and performance testing on HBase		1		
Scalable SQL and NoSQL Data Stores			1	
Solving Big Data Challenges for Enterprise Application Performance Management		1		
SQL Versus NoSQL Movement with Big Data Analytics			1	
SQL, noSQL or newSQL – comparison and applicability for Smart Spaces		1		
The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes		1		
	Totals			
	Real	Simulation	Literature Review	Other
	10	31	7	3

Figure 3 shows the relative frequency of each article category. By a large margin, most of the prior work was conducted using simulated data.

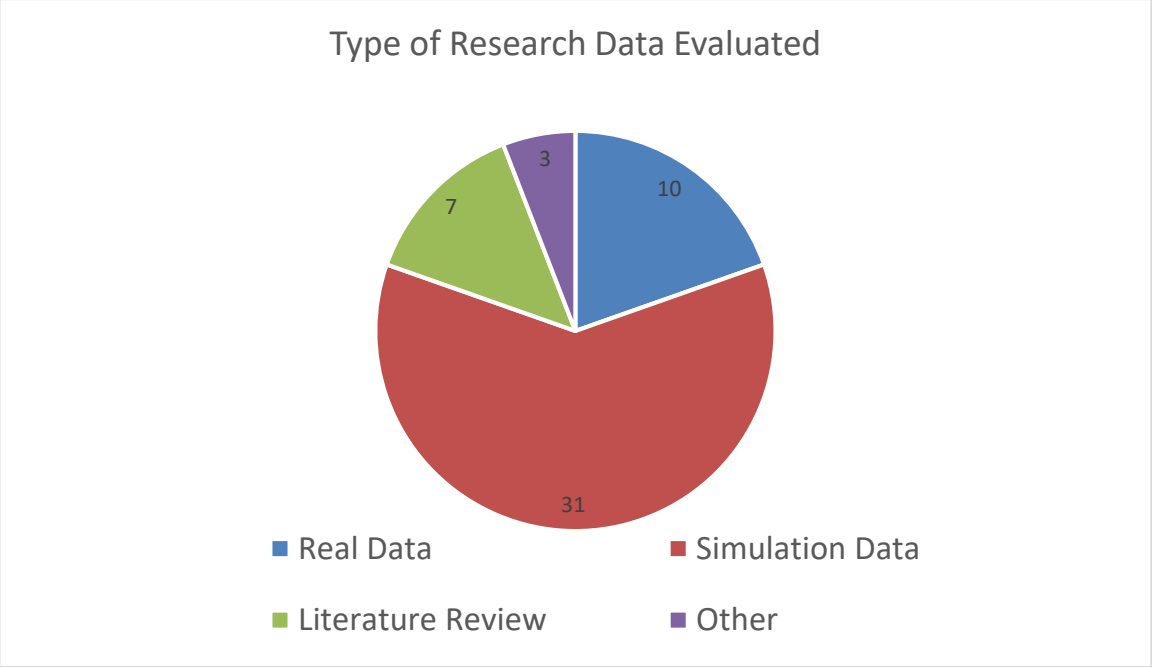

Figure 3: Method of Database Comparison

During the next step in setting up the study, it was determined which of the articles made head-to-head comparisons between different types of schemas. This is a necessary step since the literature was not consistent among which database schemas were being evaluated, so in order to connect the correlations between the five different database schemas, we needed to understand how they directly stack up against another schema. If we only looked at the overall outcomes, the frequency at which a databases schema was studied would skew the results. This required many tabulations for each database schema in direct relation to every other database schema. An example of these head-to-head tabulations converted into percentages is shown in in Table 3, where Document and Relational databases were directly compared. Percentages based on how often the

individual databases were studied were used for analysis since the total numbers would vary depending on which two schemas were in comparison. Additionally, accuracy is omitted from this table since there was not quantitative data specifically comparing Document databases against Relational databases against each other.

Table 3: Document Databases Comparison Against Relational Databases

	Query Latency	Writing Latency	Volume	Scalability
Document	67%	81%	79%	100%
Relational	33%	19%	21%	0%

Lastly, in an attempt to understand how the computing specifications affected performance characteristics in each of the database schemas, each of the articles were surveyed to see whether or not the tests performed were performed under stressed conditions or not. For our purposes, stressed means the testing tasked the computer to perform more instructions per second (IPS) than the computer was physically capable of performing. This forces the computer to operate at its peak execution rate and yields conditions that highlight the performance advantages and disadvantages. As is to be expected, most tests that were conducted under stressed conditions also included iterations under non-stressed conditions. For the purpose of this study, if results were provided under both stressed and non-stressed conditions, we used the results from the stressed conditions as the prevailing results. As shown in Figure 4, we conclusively determined 75% of the studies were performed under stressed conditions. The other 25% may have been conducted under stressed conditions, but a lack of data or computing specifications did not allow us to conclusively determine the conditions were stressed.

Since stressed testing was the desired state, if we could not conclusively determine as such, we assumed the testing was of the less desirable state of non-stressed.

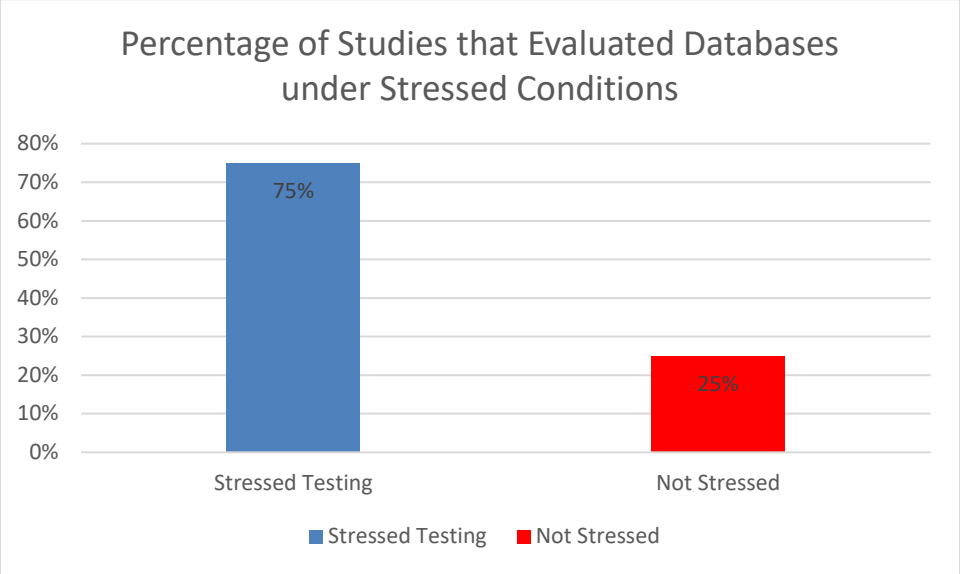

Figure 4: Percentage of Studies that Evaluated Databases under Stressed Conditions

Summary

As previously stated, the methodology used was a meta-analysis of existing research articles. The articles used provided insight into the differences in operations among the five database schemas and were further broken up to determine differences in the database performance characteristics. Additionally, we looked into what type of data the was used to generate the test results of each article. Next, we will examine the results of analyzing the data that was gathered.

IV. Analysis and Results

Chapter Overview

Upon reviewing peer reviewed articles related to the proposed research, there is a lot of information that can be gathered helping to formulate a proper starting point for the research. One common theme among most of the articles is the selection of the database type is highly dependent on the goals of the database. For example, as we learned in several of the articles, Relational databases (SQL) provide the easiest creation of databases and are easy for users to store and query data. However, there are some drawbacks among the characteristics of Relational databases as the needs change. Such as databases that will be ingesting extremely large data sets. When large data sets are introduced, the time and processing power required to operate Relational databases increases at a faster rate than that of NoSQL databases, making NoSQL a better option for any database that will fall into the category of "Big Data." (Sánchez-de-Madariaga, 2017)

Results

As previously discussed, the articles were coded based on five common performance categories (query latency, writing latency, volume, accuracy, and scalability). These categories were determined based on the articles themselves. For example, each article was coded based on the types of analysis conducted and the results were binned. After reviewing all 50 articles, the five categories naturally emerged and became the basis of comparison for this study. Figure 5 depicts the relative percentage of time each database

type was determined to be the best at a given performance category. For example, when Document databases were studied (37 out of 50 articles), 91% of the time they were the best for Query Latency, 65% of the time they were the best as Writing Latency, 48% of the time they were best at handling Large Volumes, 0% of the time they were best for Accuracy, and 8% of the time they were best for scalability. Each database time was also compared in this same manner for each performance category. Also of note, these results are only reported for observed occurrences in the literature. For example, if Key-Value databases were not compared to Relational databases on all five performance parameters, only the parameters where comparison could be identified were reported.

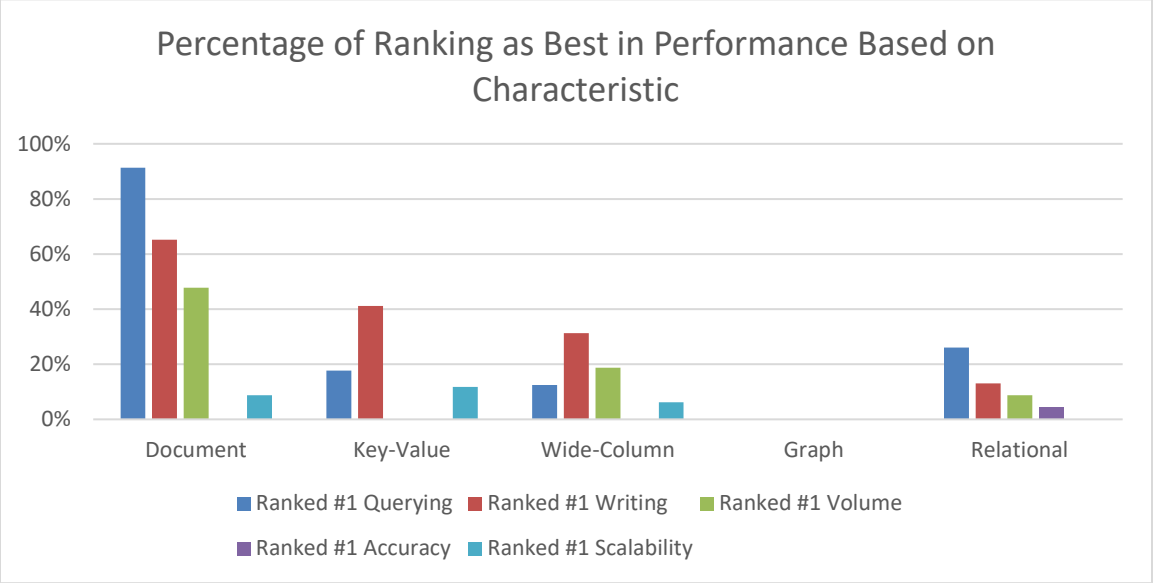

Figure 5: Percentage of Best Performance by Characteristic

While this chart is interesting and shows some relative strengths of each database type, further analysis was required to gain deeper insight into each database. Specifically of interest to this study was how well NoSQL databases performed against Relational

databases, and how well the NoSQL databases performed against each other. The following sections review the performance results for each database type.

Document Databases

Document databases were the most studied database in the literature. Figures 6 and 7 compares Document databases to Relational databases as well as to the other NoSQL databases.

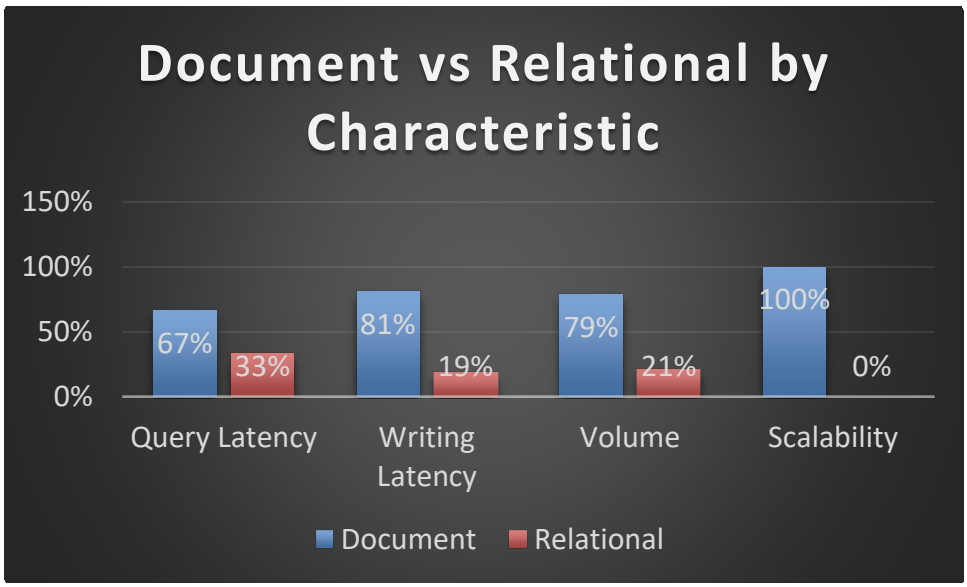

Figure 6 : Document vs Relational by Characteristic

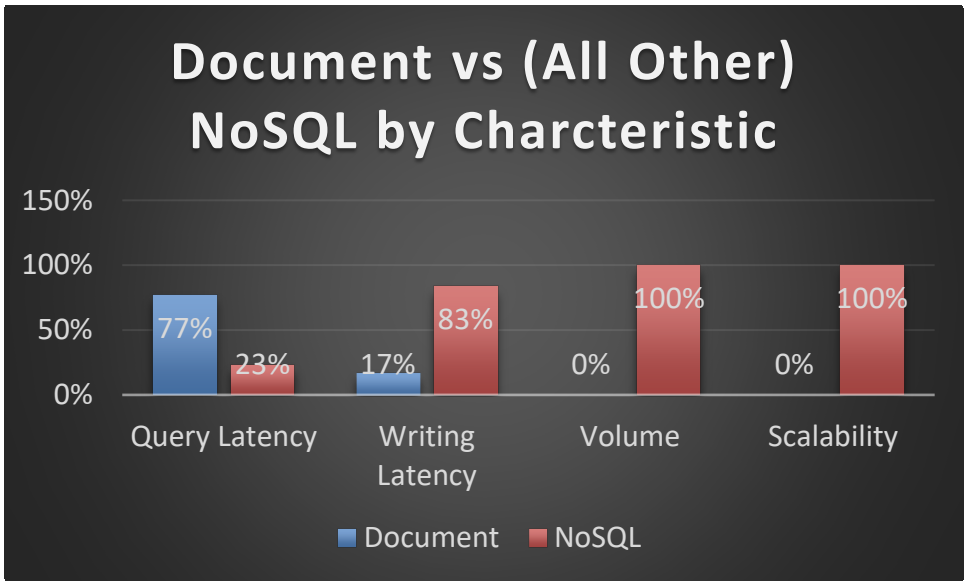

Figure 7: Document vs NoSQL by Characteristic

Document databases consistently outperformed Relational database in every category except Accuracy. When compared to the other NoSQL databases, it performed best at Query Latency 77% of the time, but on 17% of the time for Writing Latency. It was never determined to be the best at Volume or Scalability. These results indicate Document DBs excel when Query Latency is the most important consideration but may not be the best choice when other performance parameters are of greater importance.

Key-Value Databases

The next database to be analyzed was Key-Value. Figures 8 and 9 depict the results.

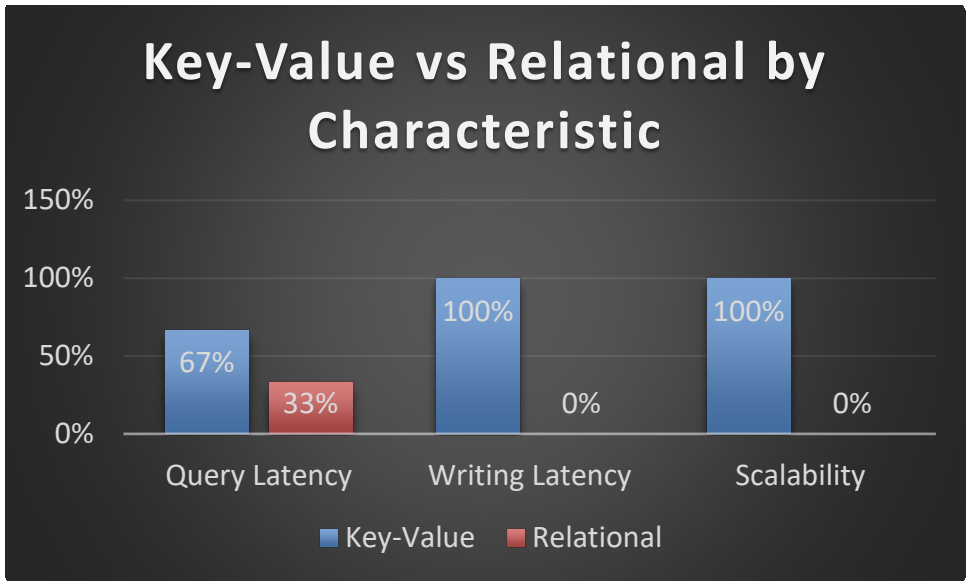

Figure 8: Key-Value vs Relational by Characteristic

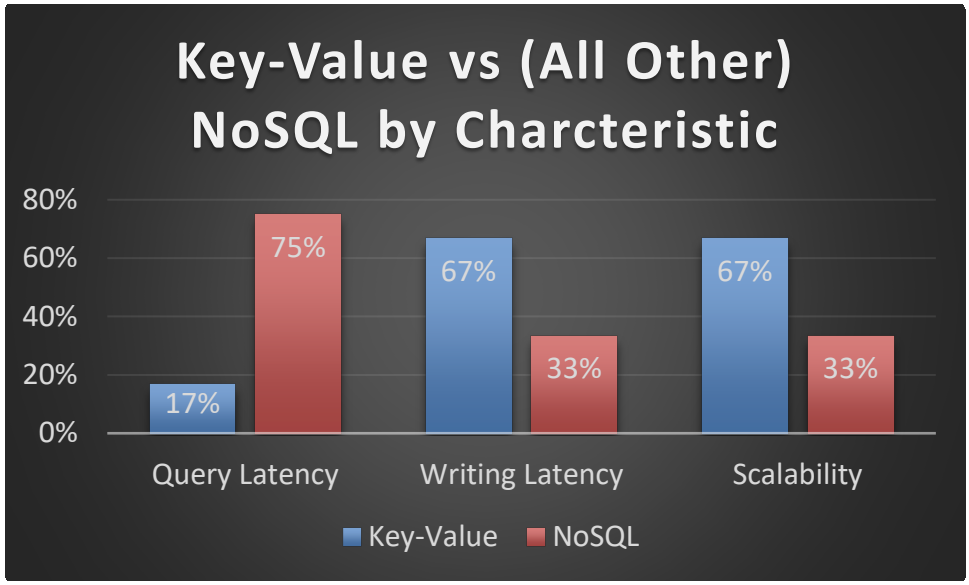

Figure 9: Key-Value vs NoSQL by Characteristic

Key-Value databases outperformed Relational databases for Query Latency, Writing Latency and Scalability. No determination could be made for Volume and Accuracy because the literature did not include results for those comparisons. When compared to

all other NoSQL databases, Key-Value was judged the best for writing (67%) and scalability (67%). However, it was only best for query latency 17% of the time.

Wide-Column

Wide-Column was the next DB analyzed. Figures 10 and 11 shows the results.

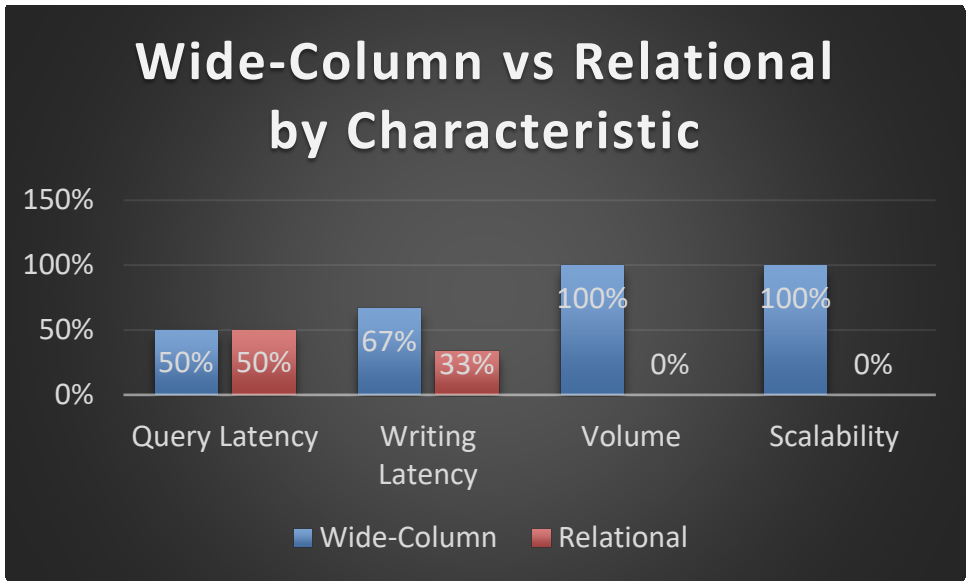

Figure 10: Wide-Column vs Relational by Characteristic

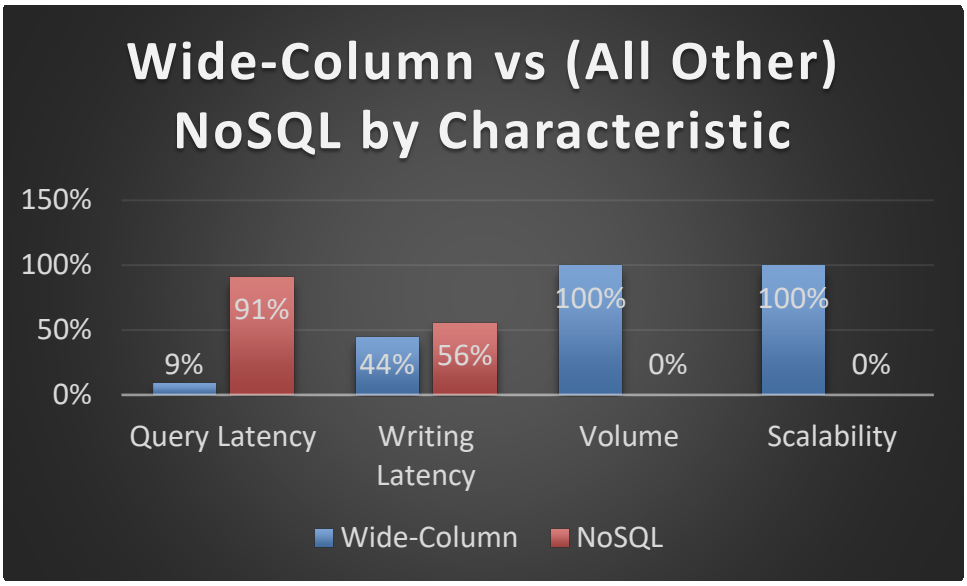

Figure 11: Wide-Column vs NoSQL by Characteristic

Compared Relational DBs, Wide-Colum performed the best at writing latency (67%), volume (100%) and scalability (100%). Compared to NoSQL, it remained the best at handling volume and scalability, while only being selected as the best at query and writing latency 9% and 44% of the time respectively.

Graph

Graph databases performed the poorest in all categories when compared to Relational databases. Similarly, it performed extremely poorly compared to NoSQL databases except for the category of accuracy. It should be noted however, that Graph databases have a different use case rather than raw performance, therefore the comparisons of only performance characteristics is a partial comparison that is not advantageous for Graph databases. Additionally, Graph databases were judged to be the best suited to ingest

previously created databases, however, this characteristic was difficult to quantify, and no other databases were compared in this manner.

Relational

The final database to be studied was Relational. This database has been the mainstay in database applications for decades. While it is unlikely it will ever be fully replaced, the results in Figure 12 indicate when query latency, volume and scalability are important, Relational databases are not the best choice.

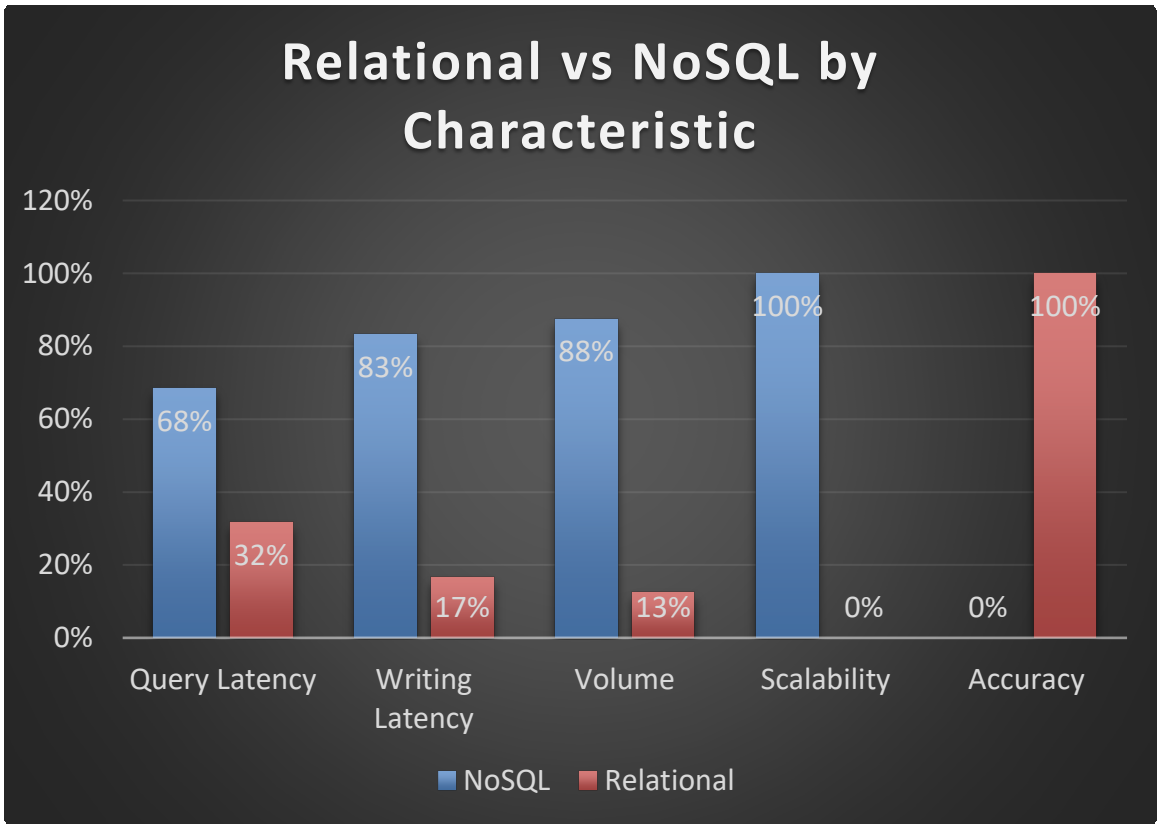

Figure 12: Relational vs NoSQL by Characteristic

Interestingly, Relational databases were judged best 13% of the time for Volume. This was counter-intuitive for volume, so a deeper look revealed this was only the case when

the definition of volume was very low and when the system was not under stress. The only category where Relational database performed better was writing latency. This result was also found to be an artifact of studies where the database was not under stress (i.e. not high volume or at large scales).

Results by Characteristic

In order to tell further describe the results of the study, a final analysis was conducted which compared "head-to-head" database types by performance parameter. The following sections report those results.

Query Latency

The first performance parameter examined was query latency. Figure 9 depicts interesting results when comparing database types directly to each other.

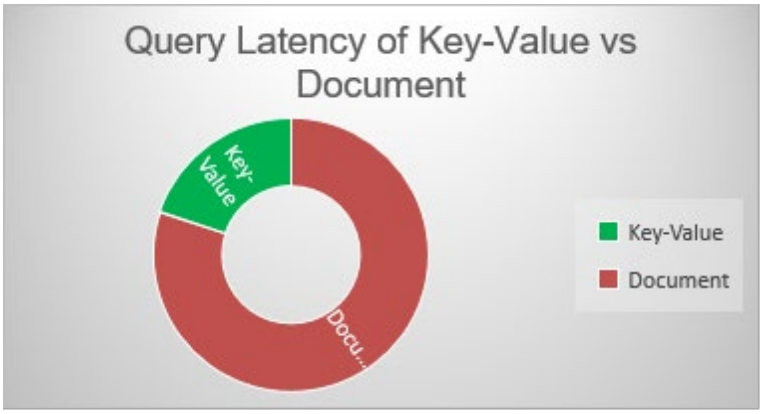

Figure 13: Query Latency Head-to-Head Comparison of Key-Value vs Document

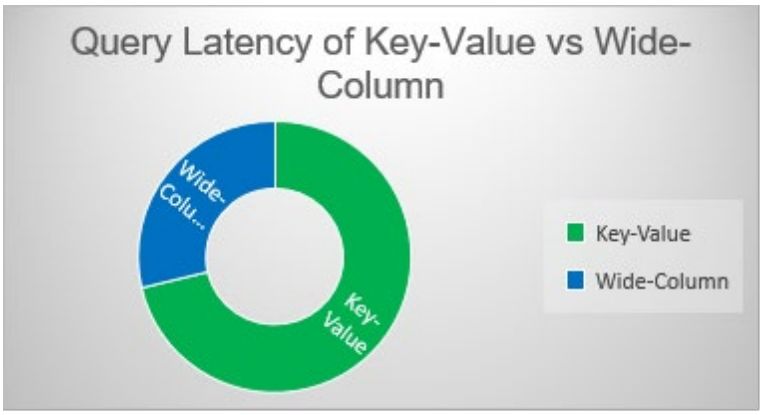

Figure 14: Query Latency Head-to-Head Compare of Wide-Column vs Document

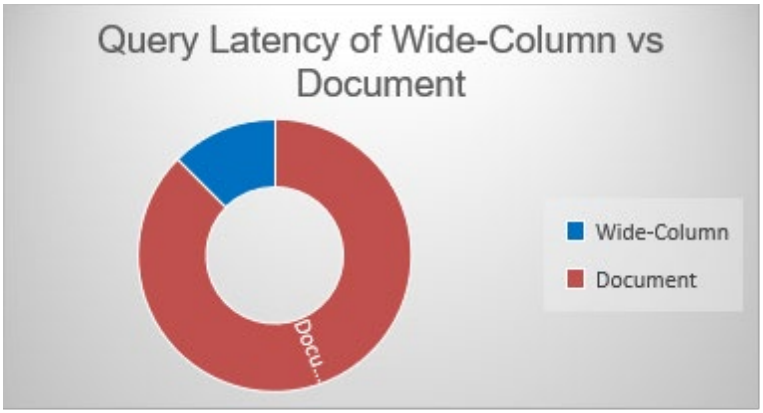

Figure 15: Query Latency Head-to-Head Comparison of Wide-Column vs Document

Querying Latency was studied in 82% of the articles making it the most common performance characteristic. Document databases overwhelmingly achieved best query performance at 91% as indicated in Figure 5. It was also the best when compared head-to-head with each database type, particularly so when compared to Key-Value and Wide-Column. However, when Key-Value was compared to Wide Column, Key-Value was the better option.

Writing Latency

For writing latency Figures 16, 17 and 18 highlight some of the results.

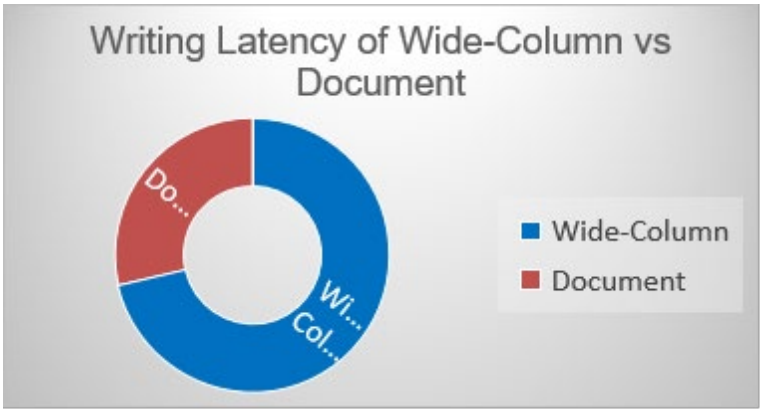

Figure 16: Writing Latency of Wide-Column vs Document

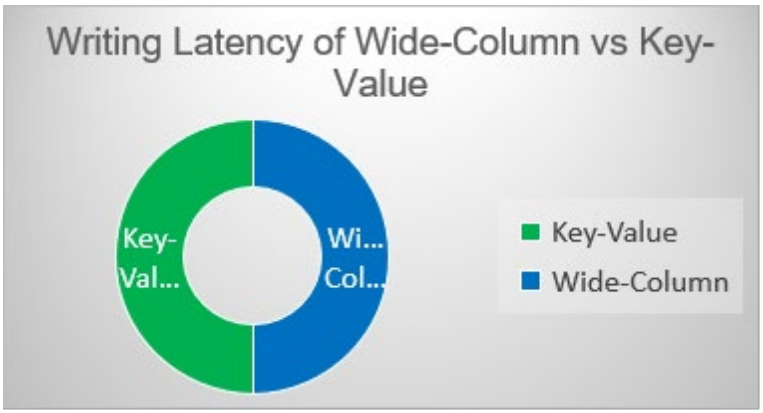

Figure 17: Writing Latency of Wide-Column vs Key-Value

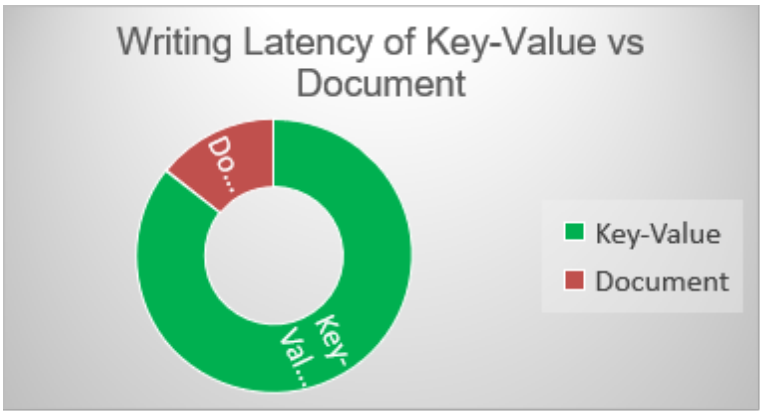

Figure 18: Writing Latency of Key-Value vs Document

Writing Latency was studied in 78% of the articles making it the second most common. Additionally, both Wide-Column and Key-Value substantially outperformed a head-to-head comparison of Document (Graph and Relational performed too poorly for a reasonable comparison with the other three). Wide-Column and Key-Value performed equally with each other in Writing capabilities.

Volume

Volume was only studied in 38% of the articles making it much less common. This performance parameter was most compared as NoSQL vs SQL rather than multiple NoSQL databases. As such, Figure 19 only shows how NoSQL performed compared to Relational databases.

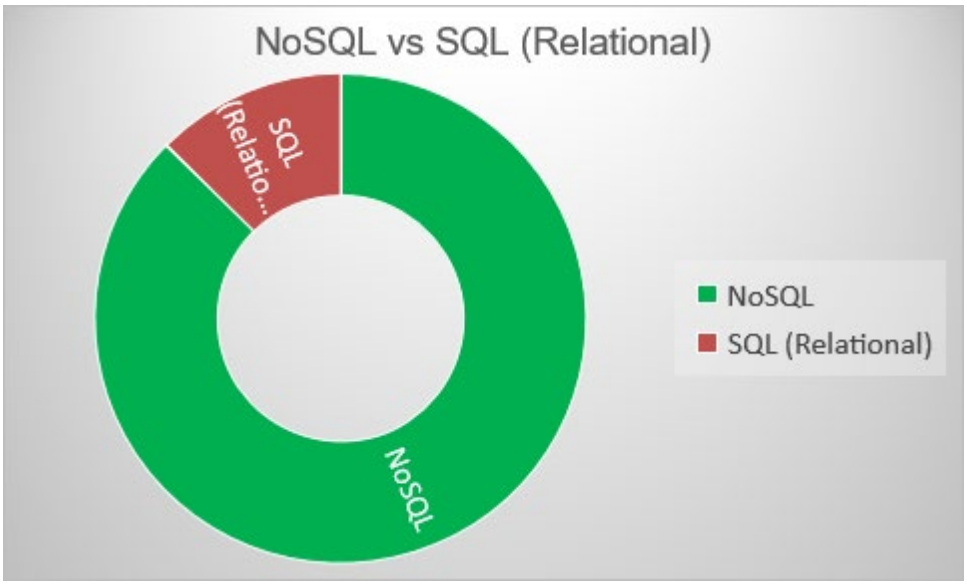

Figure 19: Head-to-Head Comparison of Volume for Relational vs NoSQL

Wide-Column performed slightly better than other NoSQL databases, but specific NoSQL databases were not compared against each other often enough in the literature for any meaningful analysis to be performed.

Accuracy

Accuracy is difficult to compare among other databases since NoSQL databases can continually update and the Accuracy will be different at separate instance in time. Therefore was only studied in 18% of the articles. As Figure 20 indicates, Relational databases performed extremely well compared to NoSQL head-to-head.

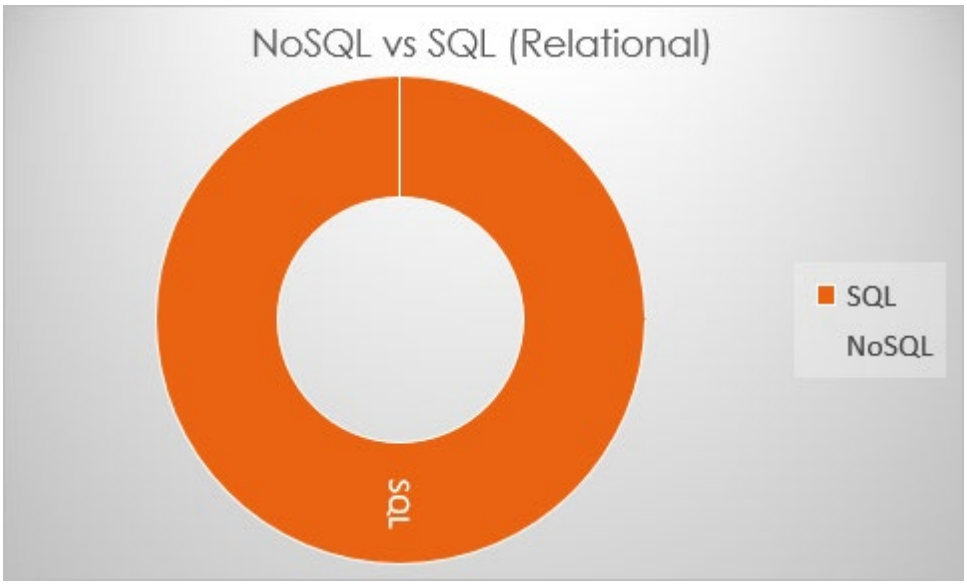

Figure 20: Head-to-Head Comparison of Accuracy in Relational and NoSQL

Scalability

The final performance parameter, Scalability, was only studied in 18% of the articles but is one of the main advantages NoSQL holds over SQL (Relational) databases as indicated in figure 21.

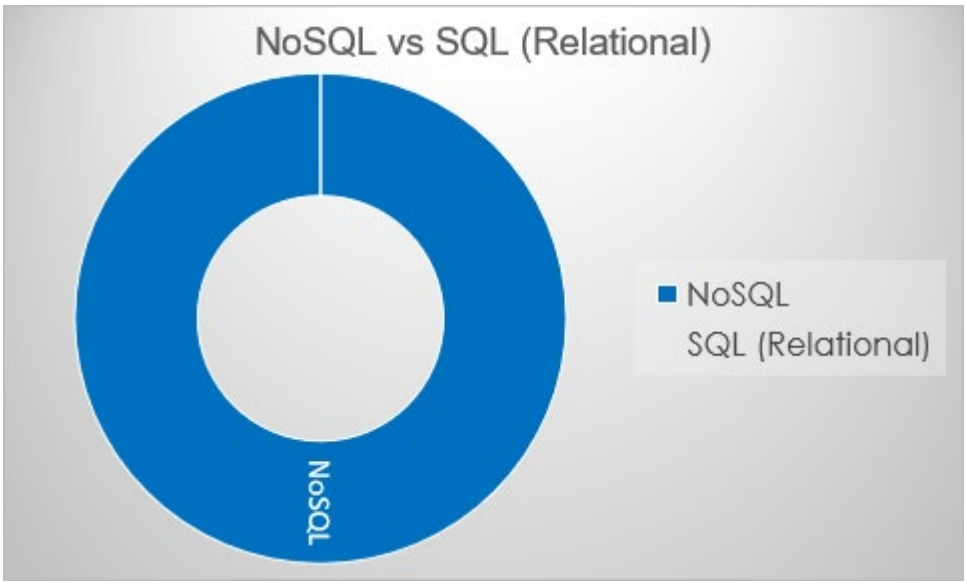

Figure 21: Head-to-Head Comparison of Scalability for Relational vs NoSQL

When comparing NoSQL head-to-head, Wide-Column performed slightly better than other NoSQL databases, but specific NoSQL databases were not compared against each other often enough on the literature for any meaningful analysis to be performed.

V. Conclusions and Recommendations

Introduction of Research

The purpose of this study was to see if any underlying trends or patterns existed between different database schemas and their relative strengths and weaknesses. The combined results of this study will help database professionals understand which database schema would be best suited for a desired characteristic. It will also guide researchers in determining the best possible methods for conducting future database research.

Conclusions

As anticipated, the specific use cases of the user will determine which database schema would be the most advantageous. However, the results of these differences were not necessarily anticipated.

Relational databases proved to provide the best accuracy and had comparable performance characteristics to NoSQL databases when they were not under stress. Therefore, Relational databases will remain a favorite schema for many years to come as databases are used every day in a manner that will never come close to stressing the systems. Take for example, practically any retail store that simply needs to track inventory in, and inventory out. This process cannot go any faster than the human in the loop executing the inventory transactions, so it is unlikely to exceed the computing abilities of the computer in use. Combine that with the simplicity and economical benefits of a Relational database, anyone who does not require extensive performance out of their

database could benefit from a Relational database. However, if there is a likely a chance the database will need to be scaled up or may operate under stressed conditions in the future, then that would likely not be the best option.

Key-Value databases yielded the best results for writing capability, suggesting that use cases with large data input would benefit from this schema. Key-Value also showed excellent performance in scalability, making them a good choice for use cases where the end application is not entirely defined. Therefore Key-Value may be an ideal choice for startup companies for two reasons, they are relatively simple so adapting to change over time is possible, and they allow for scaling if the company grows significantly in the future.

Document databases consistently outperformed other database schemas in querying latency and are one of the most popular modern database schemas. The fast querying latency makes Document databases the optimal choice when querying speed is the most important aspect and inserting new data or rewriting data do not take priority. This can be useful for applications where data analysis is an important application. Since the querying can be done quickly, it allows users to aggregate all the information for analysis in an effective manner. Common examples may include large companies with complex customer information that might benefit from a demographic analysis to better target advertising.

Wide-Column databases performed the best for volume. This type of database is best suited when there will be multiple users executing queries simultaneously. Unsurprisingly, this is the database schema is commonly used in applications with

countless users such as Google's BigTable database and would be best suited for any needs where massive amounts of data are expected.

Graph databases did not exceed all other database schemas in any of the selected performance characteristics. However, Graph databases were commonly noted as well suited to ingest previously created databases and still offer better scalability than that of a traditional Relational database. Additionally, Graph databases provide excellent user interface and can help users find relationships between data. This might be valuable for something such as online shopping where customers tend to buy complementary items. However, this aspect was not studied in this paper.

Study Limitations

In this study, we are reliant upon test cases in past research which were not set up for the explicit reason of competing against one another. Therefore, slight variations in how the databases were created could provide variations in the results. For example, Document databases utilizing XML documents appeared to operate slower than Document databases operating utilizing JSON documents. Unfortunately, due to the broad nature of this research, deep examination these two variations and how they compared with non-Document databases was out of the scope of the research.

Additionally, analysis could only be performed upon the data provided. Consequently, we could not alter test iterations to answer specific questions that may have been generated by ongoing analysis.

Recommendations for Future Research

There is still considerable research that could benefit the database schema field. For starters, more data. When we look at all the variations that can occur when setting up database performance tests, there is the possibility for considerable differences to arise in seemingly undistinguishable tests. Adding in more data from the same type of performance tests already conducted would be valuable to improve the confidence in aggregation assessments.

Furthermore, there are very few research studies conducted at looking at all the database schemas simultaneously. The lack of expansive research in this area required us to pull specific pieces of information from research to mesh up with other research, but it may not have been designed for the specific purposes of comparing the different database schemas. Research specifically designed for comparing the performance characteristics would allow for more control over the experiments performed, ideally leading to more accurate data.

Lastly, the landscape of database management systems is tied to advances in computing power and computing practices. Therefore, change is constant and at a high rate. Research will always be needed in the emerging technologies, which may include new NoSQL schemas in the near future. Additionally, databasing services are commonly moving to cloud environments with varying and complex operating practices. It is our assumption that the different database schemas would still perform in the same manner when in a cloud environment vice in-house environment, but research in that area would need to be conducted to confirm that assumption.

Summary

Before we began this study, we assumed it was advantageous to understand what needs your database will be servicing before building or selecting a database so that the proper database schema could be selected, yielding the best desired performance characteristics. After conducting the research, that assumption held true. There is not a one size fits all database schema, so understanding which one you will need it key to optimize performance.

While there certainly are common database needs that will into standard requirements, many applications will require some degree of individualized investigation to understand what schema would be best suited. For example, financial databases require complete and total accuracy at all times, therefore Relational databases are almost always going to be the best choice. However, a company requiring a database to store basic information such as customer information, operating documents, and spreadsheets, may need to decide which performance characteristics they want to maximize. If they just want to able to access the data quickly, a Document database would be the best choice. However, if the company plans to have many users trying to access the database simultaneously, then a Wide-Column schema may be best suited. Understandable, it may be difficult for the company to definitely determine what their database needs will look like in the future, but at least a basic understanding of the needs can pay dividends in the long run.

Appendix

	1st	2nd	3rd	4th
Document	91%	43%	4%	0%
Key-Value	18%	41%	12%	0%
Wide-Column	13%	19%	6%	6%
Graph	0%	0%	38%	13%
Relational	26%	52%	22%	0%

Table 4: Query Latency - Percent Rated When Compared

	1st	2nd	3rd	4th
Document	65%	57%	13%	0%
Key-Value	41%	6%	12%	0%
Wide-Column	31%	19%	13%	0%
Graph	0%	0%	25%	13%
Relational	13%	57%	4%	4%

Table 5: Writing Latency - Percent Rated When Compared

	1st	2nd	3rd	4th
Document	48%	17%	4%	0%
Key-Value	0%	0%	0%	0%
Wide-Column	19%	0%	0%	0%
Graph	0%	0%	0%	0%
Relational	9%	48%	4%	0%

Table 6: Volume - Percent Rated When Compared

	1st	2nd	3rd	4th
Document	48%	17%	4%	0%
Key-Value	0%	0%	0%	0%
Wide-Column	19%	0%	0%	0%
Graph	0%	0%	0%	0%
Relational	9%	48%	4%	0%

Table 7: Accuracy - Percent Rated When Compared

	Query Latency	Writing Latency	Volume	Scalability
Document	67%	81%	79%	100%
Relational	33%	19%	21%	0%

Table 8: Head-to-Head Comparison - Document vs Relational

	Query Latency	Writing Latency	Volume	Scalability
Document	77%	17%	0%	0%
NoSQL	23%	83%	100%	100%

Table 9: Head-to-Head Comparison - Document vs NoSQL(All NoSQL Except Document)

	Key-Value	Relational
Query Latency	67%	33%
Writing Latency	100%	0%
Scalability	100%	0%

Table 10: Head-to-Head Comparison - Key-Value vs Relational

	Key-Value	NoSQL
Query Latency	17%	75%
Writing Latency	67%	33%
Scalability	67%	33%

Table 11: Table 9: Head-to-Head Comparison - Key-Value vs NoSQL (All NoSQL Except Key-Value)

	Wide-Column	Relational
Query Latency	50%	50%
Writing Latency	67%	33%
Volume	100%	0%
Scalability	100%	0%

Table 12: Head-to-Head Comparison - Wide-Column vs Relational

	Wide-Column	NoSQL
Query Latency	9%	91%
Writing Latency	44%	56%
Volume	100%	0%
Scalability	100%	0%

Table 13: Head-to-Head Comparison - Wide-Column vs NoSQL (All NoSQL Except Wide-Column)

	Graph	Relational
Query Latency	0%	100%
Writing Latency	0%	100%
Volume	0%	100%
Scalability	0%	100%

Table 14: Head-to-Head Comparison - Graph vs Relational

	Graph	NoSQL
Query Latency	0%	100%
Writing Latency	0%	100%
Volume	0%	100%
Scalability	0%	100%

Table 15: : Head-to-Head Comparison - Graph vs NoSQL (All NoSQL Except Graph)

	NoSQL	Relational
Query Latency	68%	46%
Writing Latency	17%	83%
Volume	13%	87%
Scalability	100%	0%

Table 16: Head-to-Head Comparison - Relational vs NoSQL

Bibliography

A B M Moniruzzaman, S. A. (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. *International Journal of Database Theory and Application Vol/ 6*.

A. Gupta, S. T. (2017). NoSQL databases: Critical analysis and comparison. *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 293-299.

A. K. Samanta, B. B. (2018). Query Performance Analysis of NoSQL and Big Data. *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 237-241.

Akshay, B. S. (2019). Performance Analysis of Queries in RDBMS vs NoSQL. *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 1283-1286.

Alan Litchfield, A. A. (2017). Distributed Relational Database Performance in Cloud Computing: an Investigative Study. *Twenty-third Americas Conference on Information Systems.* Boston.

AliJaved, B. S. (2020). A benchmark study on time series clustering. *Machine Learning with Applications Vol. 1*.

Amit Kumar Dwivedi, C. S. (2012). Performance Analysis of Column Oriented Database versus Row Oriented Database. *International Journal of Computer Applications*, 31-34.

Andrea Gandini, M. G. (2014). Performance Evaluation of NoSQL Databases. *European Workshop on Performance Engineering*, 16-24.

Cattell, R. (2010). *Scalable SQL and NoSQL Data Stores.*

Ciprian-Octavian Truică, E.-S. A. (2021). The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes. *Big Data Research vol. 25*.

Colombi, J., Miller, M. E., Schneider, M., McGrogan, J., Long, D. S., & Plaga, J. (2012). Predictive mental workload modeling: implications for system design. *Journal of Systems Engineering, 15*(4), 448-460.

Dayne Hammes, H. M. (2014). Comparison of NoSQL and SQL Databases in the Cloud. *Association for Information Systems 2014 Proceedings.*

Dipina Damodaran B, S. S. (2016). Performance Evaluation of MySQL and MongoDB Database. *International Journal on Cybernetics & Informatics Vol. 5*.

Elmasri, S. N. (2016). Quantitative Analysis of Scalable NoSQL Databases. *2016 IEEE International Congress on Big Data (BigData Congress)*, 323-326.

Fan, E. T. (2016). Performance Comparison between Five NoSQL Databases. *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, (pp. 105-109).

Francisco L. Loaiza-Lemos, D. V. (2017). *Assessment of Graph Databases as a Viable Materiel Solution for the Army's Dynamic Force Structure (DFS) Portal Implementation: Part 1, Preliminary Characterization of Data Sources, Representation Options, Test Scenarios and Objective Metrics.* Alexandria, Virginia: Institute for Defense Analysis.

Google. (2021, May). *Overview of Bigtable.* Retrieved from cloud.google.com: https://cloud.google.com/bigtable/docs/overview

Gourav Bathla, R. R. (2018). Comparative study of NoSQL databases for big data storage. *International Journal of Engineering & Technology vol. 7*.

Green, J. W. (2008). *A Comparison of the Relative Performance of XML and SQL Databases in the Context of the Grid-SAFE Project.* The University of Edinburg.

Han-Sheng Huang, S.-H. H.-W. (2015). Load balancing for hybrid NoSQL database management systems. *Proceedings of the 2015 Conference on research in adaptive and convergent systems*, (pp. 80-85).

Henricsson, R. (2011). *Document Oriented NoSQL Database, A Comparison in MongoDB and CouchDB using a Pythin interface.* Bleking Institute of Technology.

Hossain, A. B. (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. *International Journal of Database Theory and Application*.

Hurbungs, S. R. (2020). Evaluating the performance of SQL and NoSQL databases in an IoT environment. *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*, 229-234.

Jacob Berlin, A. M. (2002). Database Schema Matching Using Machine Learning with Feature Selection. *International Conference on Advanced Information Systems Engineering*, 452-466.

K. E. Roopak, K. S. (2013). Performance Comparison of Relational Database with Object Database (DB4o). *2013 5th International Conference and Computational Intelligence and Communication Networks*, 512-515.

K. Mahmood, K. O. (2019). Comparison of NoSQL Datastores for Large Scale Data Stream Log Analytics. *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, 478-480.

Kolonko, K. (2018). *Performance comparison of the most popular relational and non-relational database management systems.* Karlskrona, Sweden: Blekinge Institute of Technology.

Kumaran, S. P. (2019). A Quantitative Performance Analysis between Mongodb and Oracle NoSQL. *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, 387-391.

Lourenço, J. C. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*.

M. Jung, S. Y. (2015). A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment. *2015 8th International Conference on Database Theory and Application (DTA)*, 14-17.

Manoharan, Y. L. (2013). A performance comparison of SQL and NoSQL databases. *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 15-19.

Matteo Gabetta, I. L. (2015). BigQ: a NoSQL based framework to handle genomic variants in i2b2. *BMC Bioinformatics vol. 16*, 415.

P. Bagade, A. C. (2012). Designing performance monitoring tool for NoSQL Cassandra distributed database. *International Conference on Education and e-Learning Innovations*, 1-5.

Puntheeranurak, W. P. (2017). A comparative study of relational database and key-value database for big data applications. *2017 International Electrical Engineering Congress (iEECON)*, 1-4.

R. Pasumarti, R. B. (2017). Capacity Measurement and Planning for NoSQL Databases. *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, 390-394.

Rani, S. G. (2016). A comparative study of elasticsearch and CouchDB document oriented databases. *2016 International Conference on Inventive Computation Technologies (ICICT)*, 1-4.

Ricardo Sánchez-de-Madariaga, A. M.-R.-B. (2017). Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NoSQL approaches. *BMC Medical Informatics and Decision Making vol. 17*, Article 123.

Rudolf, C. (2007). SQL, noSQL or newSQL – comparison and applicability for Smart Spaces. *Proceedings of the Seminars Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM)* (pp. 39-46). Munich, Germany: Technical University of Munich.

S. Chickerur, A. G. (2015). Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications. *2015 8th International Conference on Advanced Software Engineering & Its Applications (ASEA)*, 41-47.

Sánchez-de-Madariaga, R. M.-R. (2017, August 18). Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NoSQL approaches. *BMC Medical Informatics and Decision Making volume 17*.

Sarah H. Kamal, H. H. (2019). A Qualitative Comparison of NoSQL Data Stores. *International Journal of Advanced Computer Science and Applications vol. 10*, 330-338.

Sergio Miranda Freire, D. T.-K. (2016). Comparing the Performance of NoSQL Approaches for Managing Archetype-Based Electronic Health Record Data. *Plos One*. Retrieved from Plos One.

Sitalakshmi Venkatraman, K. F. (2016). SQL Versus NoSQL Movement with Big Data Analytics. *I.J. Information Technology and Computer Science*, 59-66.

Sudhir Mudur, A. M. (2013). NoSQL databases: MongoDB vs cassandra. *C3S2E '13: Proceedings of the International C* Conference on Computer Science and Software Engineering*, (pp. 14-22).

Tilmann Rabl, M. S.-A.-V.-M. (2012). Solving Big Data Challenges for Enterprise Application Performance Management. *Proceedings of the VLDB Endowment Vol. 5.* Istanbul, Turkey.

Vansteenberghe, S. J. (2013). An Empirical Comparison of Graph Databases. *2013 International Conference on Social Computing*, 708-715.

Veronika Abramova, J. B. (2014). Experimental Evaluation of NoSQL Databases. *International Journal of Database Management Systems Vol. 6*.

Wajid Ali, M. U. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. *Asian Journal of Research in Computer Science*, 1-10.

Wasnik, H. F. (2016). omparison of SQL, NoSQL and NewSQL databases for internet of things. *2016 IEEE Bombay Section Symposium (IBSS)*, 1-6.

Wei, W. N. (2013). Review of NoSQL databases and performance testing on HBase. *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, 2304-2309.

Xiaoming Wang, C. W. (2019, January). Big data management challenges in health research—a literature review. *Briefings in Bioinformatics*, pp. 156-167.

Yusuf Abubakar, T. S. (2014). Performance Evaluation of NoSQL Systems Using YCSB in a resource Austere Environment. *International Journal of Applied Information Systems Vol. 7*, No. 8.

Zachary Parker, S. P. (2013). Comparing NoSQL MongoDB to an SQL DB. *Proceedings of the 51st ACM Southeast Conference.* ACM.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	
11-29-2021	Master's Thesis	December 20202 – December 2021

TITLE AND SUBTITLE

META-ANALYSIS OF PERFORMANCE CHARACTERISTICS OF MODERN DATABASE SCHEMAS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Grove, Carter, GG-12, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-MS-D-047

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research
875 North Randolph Street, Suite 325
703-696-7797

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRUBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Industry and academia alike are more commonly using databases as solutions to advanced and complex problems. Unfortunately, not all database schemas are created equal and can yield different advantages in different areas. To try to understand what database schema might be best suited for a user's needs, we sought out to distinguish how databases are measured against each other, what their performance characteristics are, and what advantages each type of database inherently possesses. To allow for the ingestion of data across the five different categories of database schemas, we used a met-analysis of past literature and aggregated the data to form the basis of data to analyze. The data was then used to compare which database schemas exhibited the best performance for accuracy, scalability, transactions, query latency, and writing latency.

After analyzing the data, a mix of NoSQL databases performed the best for scalability, transactions, and query and writing latency, making them advantageous for database solutions for unique problems. Relational databases maintained the best accuracy among databases and are the cheapest solution, making them suitable for basic databasing needs. Most importantly, many applications will require some degree of individualized investigation to understand what schema would be best suited.

15. SUBJECT TERMS

Database, SQL, NoSQL, Document, Key-Value, Wide-Column, Graph, Relational

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. B. Langhals, AFIT/ENV
U	U	U	UU	19	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 7402 (brent.langhals@afit.edu)