Available online at www.garmian.edu.krd

Passer Journal

Passer 4 (2022) 45- 57

http://passer.garmian.edu.krd/

A Survey of Comparing Different Cloud Database Performance: SQL and NoSQL

Twana Hussein Shareef[1*], Karzan Hussein Shareef [2], Bilal Najmaddin Rashid[3]

[1]*Department of Information Technology, College of Commerce, University of Sulaimani, Sulaimani, Kurdistan Region, Iraq*
[2]*Software engineering Department, Faculty of engineering and computer science, Qaiwan International University-UTM Franchise, Sulaimani, Kurdistan Region, Iraq*
[3]*Department of Information Technology, College of Commerce, University of Sulaimani, Sulaimani, Kurdistan Region, Iraq*

Received 05 November 2021; revised 01 January 2022,
accepted 15 January 2022; available online 15 February 2022

doi:10.24271/psr.40

ABSTRACT

The scales in databases have recently increased, and more expansion is expected in the future. The storage costs gradually have slowed down and quickly expanded the storage capacity. Cloud intrusion has modified the comparisons in recent years. Database performance plays an essential part in the market. Cloud Database focuses on new, conventional databases, specific applications for scalability, dynamic devices, and ease of use. Cloud databases are primarily used for data storage, retrieval, modification, and analysis by tools like business intelligence. These tools can build new business strategies and demonstrate scalability and elasticity while managing vast amounts of data with reliable, customized, and cost-effective services in various applications. This paper provides an overview of cloud computing, Cloud database architecture and kinds, and database as a service. It also highlights the characteristics, deployment, and service model of cloud computing and the performance and functionality of the various SQL and NoSQL cloud database applications and services required to evaluate them. It focuses on the different parameters to assess their performance, such as ease of software portability, transaction capabilities, and the maximum amount of data stored. The primary purpose of this paper is to assist businesses and individuals in understanding how cloud computing may provide them with dependable, personalized, and cost-effective services in many applications.

Keywords: Cloud Computing, Cloud database, DaaS, SQL, NoSQL, Amazon RDS

1. Introduction

In recent years, there has been a spectacular transformation in the IT industry regarding commercial applications. E-apps have replaced or transferred programs previously hosted on a single server in the organization's IT architecture. In addition, system storage has taken the position of specialized storage. The pay-per-use model, flexibility, and lower cost are the key reasons why distributed computing has become a reality. Cloud databases are now being viewed as a solution for programmers, designers, and architects that need to store information for their applications in a flexible and highly accessible manner from the backend when needed[1] .

Businesses are migrating their software to the Cloud. Despite obstacles such as a shift in the economy (data location restrictions and security), costs and flexibility will prevail sooner or later, as even national security agencies dedicated to providing hosting cloud operations demonstrate[2]. A cloud provider that sells

* Corresponding author
E-mail address: twana.shareef@univsul.edu.iq (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

DBMS as-a-service offers many benefits, from streamlined software updates, node failure tolerance, and hot disk management to the architecture that "decouple(s) s storage tier from the compute tier." It also facilitates network traffic reductions by moving DBMS features as the log applicator to the storage stage[3].

According to the researcher[6], a database in the Cloud is a mixture of various site collections (known as nodes). Each site collection is connected collectively in its database via a communications network. Many companies want to move their applications to the Cloud to take advantage of cloud computing; hence cloud database systems are a new research topic. Whether conventional or Cloud, businesses evaluate the success factor in databases regardless of the paradigm. According to[4], Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) are services provided by cloud computing. Also, Database-as-a-Service enables enterprises to set up cloud computing databases and gives users access to tasks such as the store, retrieval, updating, and deletion of data. According to[5], Data as a service (DaaS) and database as a Service (DbaaS) are paradigms for delivering data and database management systems (DBMS) on demand. The (DBaaS) concept is gaining traction in the Cloud. Organizations adopt this paradigm to achieve higher security, higher availability, and lower and more flexible costs while maintaining high performance. However, it has become evident that traditional, monolithic database architecture cannot meet these objectives in the Cloud[6]. Governments, institutions, and businesses most commonly use DaaS to make data (expenses, budgets, economic or census data) accessible to public or private users over the Internet. Despite their similarity, DaaS and DBaaS are not the same things[5]. This paper addressed cloud computing's characteristics, deployment, and service model, as well as the performance and usefulness of the numerous SQL and NoSQL cloud database applications and services that must be evaluated. It focuses on various factors to assess their performance, such as software portability, transaction capabilities, and the maximum amount of data stored. The primary goal is to help companies and individuals understand how cloud computing can provide trustworthy, personalized, and cost-effective services in various applications. Figure 1 shows an overview of cloud computing.

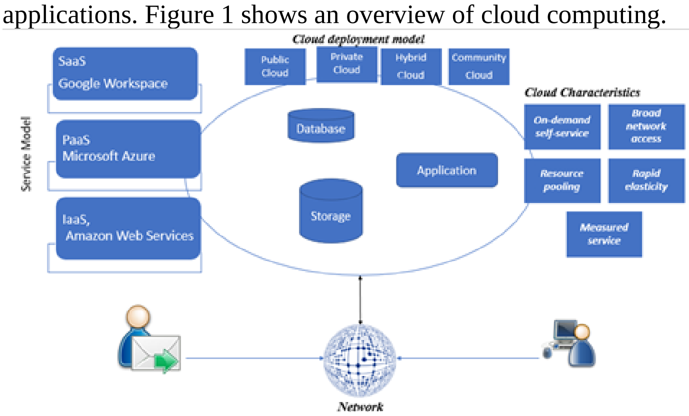

Figure 1: Overview of the cloud computing.

A review of the recent related work is given in Section 2. Section 3 offers a description of the common cloud database types. Also, the paper explains general cloud computing concepts in section 4. Also, section 5 provides an overview of the cloud database-as-a-service. The following is how the rest of the paper is organized: Section 6 explains the Cloud database architecture. Section 7 gives some examples of cloud database applications. While Section 8 offers a straightforward way to compare and evaluate cloud databases. Section 9 explains the roles of DBAs in the cloud Section 10 provides a discussion. Finally, in Section 11, a conclusion of the paper is presented and some recommendations are presented based on the evaluation section.

2. Literature Review

Different databases are needed to make a valid and robust comparison case, same as how the experiments[7] follow. Other cloud database systems and various cloud providers can cover the experimental base. As they have seen in their preliminary research on this topic, the unique data from other suppliers can significantly influence the measured variance and frequency of anomalies. Furthermore, the advantages and disadvantages compared by[8] focused on the three most important aspects of cloud computing and their issues: support, performance, control, and security. Furthermore, data must be handled correctly in the cloud environment, and ACID-compliant RDBMS procedures should be employed for proper transaction management and accuracy. ACID (Atomic, Consistent, Isolated, and Durable) transaction attributes are guaranteed by relational databases. However, the CAP theorem implies that only two of the three properties (consistency, availability, and partition tolerance) can be attained simultaneously in distributed contexts. RDBMS excel at consistency and availability, but they struggle to scale. NoSQL databases' primary idea is to relax one of these two criteria, namely Consistency and Availability, to improve scalability. In contrast to ACID, they give what is known as a BASE (Basically Available, Soft State, and Eventually Consistent) qualities. Which of the two features NoSQL database systems loosen, and how much they reduce it, varies. On the other hand, many provide eventual consistency to maintain high scalability and availability[9].

Mirage Image Management System, Client Based Privacy Manager, Transparent Cloud Protection System, and Secure and Efficient Access to Outsourced Data are some of the current ways for encrypting data. However, cloud service providers still have problems, such as safeguarding user information (occasionally, data is successfully encrypted, but the decryption is not possible due to critical losses)[10]. Continuing the article done by[11] Introduced simple cloud database information and principles and described the essential characteristics.

This paper has gathered many cloud databases, compared them in our tables that cover many categories, and chose the best ones for big and small organizations. We have also discussed the challenges and advantages of cloud databases and how they can be utilized significantly. In our comparisons, the reader can learn which of the many famous DaaS are best suited for them.

3. Cloud Database Types

Carlo Strozzi first used the term NoSQL (Not Only SQL) in 1998, but a non-relational approach was also used to store and retrieve unstructured data quickly, unlike relational databases, which apply mathematical relationships between tables. Storing data in non-relational does not have a clear and well-structured mechanism to join data from a diverse table structure to one another; this is called accessible schema architecture. Most NoSQL databases are distributed, horizontally scalable, and open-source—moreover, several Non-relational database management systems such as MongoDB, Cassandra, OrientDB, and Aerospace. The performance is a significant part of both types of databases (Relational and non-relational databases) to store colossal data rapidly and quickly access the data[12]. The structure of both types is shown in figure 2.

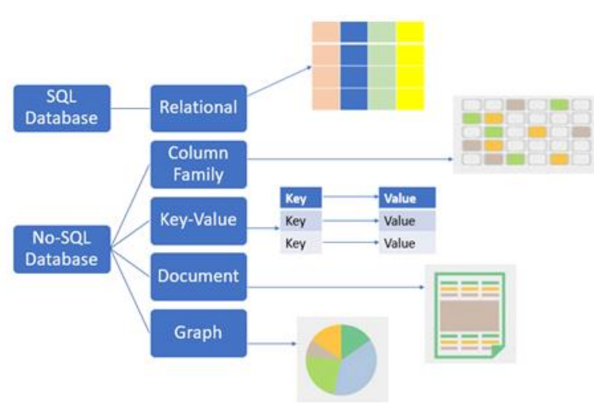

Figure 2: SQL and NoSQL Structure.

Databases are divided into two categories: NoSQL and SQL.

- SQL is a database language (stands for Structured Query Language), Developed for most databases as a high-level standard interface, generally used for DDL and DML relational system management (RDBMS). Each recognizes SQL as the questionnaire language in the relation template databases. The MySQL, MS-SQL Server, and Oracle Databases. For SQL, there is a table-based database. Data is stored in fixed-row, column tables in the table database Table[13].

- **NoSQL**, not only SQL, is a non-relational database management system. MongoDB, Cassandra, CouchDB, and HBase are the most popular NoSQL databases. In addition, NoSQL Top online corporations such as Amazon, Google, and LinkedIn have pioneered and expanded databases a lot with the current popularity of "Big Data." The primary distinction between the non-relation data model and the conventional model is that the non-relation model is built with a relatively low accuracy requirement to process an enormous amount of data in the second. Consequently, it relaxes the ACID constraints provided by many relational database systems to improve performance[13]. NoSQL has four types of databases[13]:

- **Document Stored**: Document-oriented database definition is the central concept. One of the most commonly used document-based databases, MongoDB, is the JSON format; others such as BaseX or YAMLDB are also users of XML and YAML storage.

- **Key-Value Stored:** The most easy-to-understand database of these four is Key-value. The database management system saves knowledge (or hash table). BerkeleyDB, LevelDB, and Redis are common central value databases.

- **Column Stored**: A column-driven database stores tables by columns instead of rows. This distinction is mainly controlled by the database management systems, which usually allow a column-oriented database to be compatible with conventional row-based databases; they both can use SQL to load and query data. Some examples are SAP HANA, Amazon Redshift, and Sybase IQ, column-orientated databases.

- **Graph stored**: A graph database is a database that stores information with nodes, corners, and characteristics. Nodes are individuals or corporations, whereas edges and assets are connections amongst entities. We can model and store data naturally on a logical level with graphs. AllegroGraph and gStore are samples of graph databases.

4. Cloud computing services

In industry, there has been much debate about Cloud computing definition. Cloud computing is a term that tends to be derived from diagrams of the internet network as a cloud. However, the US National Institute of Standards and Technology (NIST) has drawn up a working description covering the generally accepted aspects of cloud computing. The NIST defines cloud computing as a model that enables convenient, on-demand network access to a pool of adjustable computing resources[14].

A. Cloud characteristics

The NIST description for cloud computing is among the well-defined cloud computing descriptions and is frequently cited in documents and projects of the United States government. This description defines five core cloud computing characteristics. The core characteristics are[15]:

- ***On-demand self-service***: Computing resources can be obtained and used at any time, even if there is no human contact with cloud service providers. Power processing, storage, and virtual machines are examples of resources.
- ***Broad network access***: Heterogeneous gadgets like laptops or cell phones may connect to previously listed resources.
- ***Resource pooling***: Cloud services providers pool their resources, which many users share. For instance, a physical server may host multiple virtual machines from various users, also recognized as multi-tenancy.
- ***Rapid elasticity***: A user can quickly gain more cloud resources by scaling out. They can scale it down when they are no longer needed by releasing these tools.
- ***Measured service***: The use of resources is measured employing appropriate methods such as uses of storage monitoring, CPU time, and bandwidth.

B. Cloud service models***:*** The above section clarified the functionality for all clouds. Nonetheless, each Cloud provides users with a different level of abstraction, which is called a service model in the definition of NIST. The three most popular models of service are[15]:

- ***Software as a Service (SaaS):*** A consumer used a web browser to access applications created by others and provided a Web service. Users have no power or access to the underlying infrastructure used to host the applications at the SaaS level. For example, software3 and Google Docs4 are typical examples of Salesforce customer relationship management.
- ***Platform as a Service (PaaS***): Apps are created utilizing programming languages and tools supplied by the PaaS provider in this situation. PaaS provides consumers with the highest level of abstraction, allowing them to focus on their application rather than the underlying infrastructure. Users manage or access the underlying infrastructure needed to host applications at the PaaS level, much as with SaaS. Google App Engine5 and Microsoft Azure6 are two famous PaaS examples.
- ***Infrastructure as a Service (IaaS***): This is where users procure computer resources from an IaaS provider like process power, memory, and storage to launch and run their applications from resources. In contrast to the PaaS model, the IaaS model is a low abstraction level that enables users to use virtual machines to connect to the underlying infrastructure. It helps users to launch any software stack above their operating system. IaaS gives users more flexibility than PaaS. However, it offers flexibility with prices, and the operating system is upgraded and patched at the IaaS level by users[16]. EC2 and S3are common examples of IaaS from Amazon Web Services. Figure 3 explains what providers and users manage in IaaS, PaaS, and SaaS.

IaaS	PaaS	SaaS
Application	Application	Application
Data	Data	Data
Runtime	Runtime	Runtime
Middleware	Middleware	Middleware
OS	OS	OS
Virtualization	Virtualization	Virtualization
Servers	Servers	Servers
Storage	Storage	Storage
Networking	Networking	Networking

What Providers Manage	What Users Manage

Figure 3: Providers and users manage in (IaaS, PaaS, and SaaS).

C. Cloud deployment model

Clouds use the NIST definition services, but clouds depend on who owns and operates them. The four standard models are[15]:

- ***Private Cloud:*** A cloud that any enterprise uses. The Cloud might be managed by the company or by a third party. Private cloud organizations include the St Andrews Cloud Computing Co-Laboratory8 and Concur Technologies[17].
- ***Public Cloud***: A cloud that can be used by the general public (for a fee). Large corporations such as Microsoft, Google, or Amazon often control public clouds and require significant investment.
- ***Community cloud:*** Many businesses share a cloud that is often tailored to their specific requirements. The Open Cirrus cloud test can be thought of as a collective cloud that encourages cloud science[18].
- ***Hybrid Cloud***: A combination of three deployment models creates the Cloud. Each Cloud in a hybrid cloud can be managed separately, but apps and data can migrate through the hybrid Cloud. Hybrid clouds permit cloud explosion, enabling a private cloud to explode into the public Cloud if more resources are needed. Furthermore, there is one more cloud service called Database-as-a-Service (DaaS). Organizations use this service to host their databases in the Cloud, and the next section looks at how to use it (DaaS).

5. Cloud Database-As-Service

Customers can request access to a database hosted by the cloud service provider over the Internet. Cloud computing is used to create the cloud database, which means that cloud service providers employ their own software and infrastructure. Cloud computing can be described as a new era in IT in terms of cost savings and improved application performance. This trend indicates that organizations will soon be focusing on cloud apps. The majority of cloud databases are utilized as a service. It's also known as (DBaaS). In many enterprises around the world, the cloud database will be the most advanced technology for managing massive datasets. However, it's not easy to use a relational database and distribute it over a cloud server.

Furthermore, it implies that additional nodes are brought online when needed, increasing database performance. Data must be sent through multiple data centers located in different countries. The database must be accessible at all times so that users can retrieve data when they need it[19]. After a network breakdown, Cloud Computing immediately retrieves the information. Cloud storage differs from the relational database management system in the cloud structure[19] because it comprises data from multiple data centers in various locations. A cloud database includes several nodes set for query services and data centers in different geological areas and corporate data centers. This connection is required for cloud services to provide easy and complete access to the database. The database can be accessed via cloud services in a variety of ways; for example, the user can connect via the web or a phone and access the cloud database via 3G or 4G services[21]. To better characterize the cloud database function, we shall propose a business intelligence framework. Businesses use BI apps to store enormous volumes of data, and their customers use them to store data.

We assume that the user accesses the cloud database over the Internet. The Internet links data centers, cloud data centers, and data users[19]. Peer-to-peer communication is ideal for this purpose. The goal of peer-to-peer communication is for a single node to handle any request made by the user; it may appear hard, but this type of node system is simple to solve using the data map of each node in each cloud database. This data map allows users rapid access to detailed information.

6. Cloud Database Architecture

The advancement of cloud computing systems is difficult, such as the availability of a service, data privacy. An effective cloud database management system has been designed to fulfill the following objectives: availability, scalability, elasticity, multiple loads, high availability, and the ability to work in a diverse environment[22]. A Federalized Cloud Environment is required to strengthen the complexities and achieve the goals of Cloud, and five layers with their goals and challenges are described[23]:

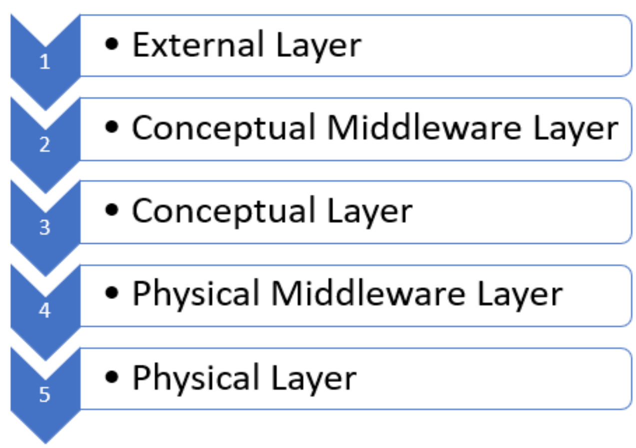

Figure 4: Layered Architecture of Cloud Database.

6.1 External Layer:

The users can view this layer. More and more companies switch their data storage software to the inexpensive commodity cloud from pricey, high-end servers—the service provider's key role in controlling and executing the services with total security and accountability.

6.1.1 *Manageability*:

It is the management of the different users in this layer. It also documents the CDMBS (Cloud Database Management System) used by a specific person, the time for user payment. This layer also retains a user's payment status. This layer can also produce reports, such as a user's payment history, using a user history. Finally, this layer can pass power to the next level once the user's credibility and legitimacy have been tested.

6.1.2 *Security*:

It uses user IDs and passwords, and the external layer offers user authentication. User authentication makes it possible to recognize someone or something as a legal one. Legal access to the database system in the Cloud is allowed, but unauthorized access is prohibited.

Transparency: The primary function of this layer is to provide clarity to the users. Transparency is one of the advantages of cloud databases, which means that users are not familiar with the physical location of data. Transparency makes it easier to create different types of real-time applications. Its versatility helps mobility and more.

6.2 Conceptual Middleware Layer:

This layer allows conceptual heterogeneity to be concealed between various databases such as SQL, DB2, Oracle, and more. The interoperability of this layer. Interoperability ensures that their underlying databases work independently. For example, if a client in the 'A' database wishes to share the data with a different client in the 'B' database, they can share the data using this layer irrespective of the various databases underlying that client.

6.3 Conceptual Layer:

This layer reflects the entire database's logical structure and deals with internal data processing. Since Cloud manages different types of data, users can merge the user's conventional data with data on the Cloud to provide all of these functions with various systems. Some systems or languages have been built for Cloud, including Bigintegrator, SQLMR, Salesforce, and Object query language, which gives username results based on their research so that consumers are unaware of the factors underlying their tasks. This layer deals with these problems below-:

6.3.1 *Programming Techniques*:

This layer resolves issues that are ideally suited for Cloud databases, for example, query language or programming techniques.

6.3.2 *Query Processing and Optimization*:

This layer is in charge of delivering the results with minimum costs in a short period. It may use Optimizer for this query. Query Optimizer will search for the least expensive implementation proposal among the many other proposals that respond in the same implementing method.

6.3.3 Security:

The key concern at this layer is that no unauthorized user can kill, change or copy information. The conceptual layer includes authorization control methods that prohibit unauthorized users from accessing sensitive information. In addition, multilevel protection policies are included.

6.4 Physical Middleware Layer:

This layer allows heterogeneity to be concealed across various platforms such as Windows, Mac OS, and Linux. Interoperability: Interoperability means that this layer operates independently of its underlying platforms. Customers who use Mac OS cannot access data from another customer's base, which is challenging to use the Windows OS.

6.5 Physical Layer:

The backend of a cloud database is monitored by a Physical layer, which constantly monitors and configures the database to ensure optimal cloud scaling, higher availability, multitenancy, and resource allocation. The issues with this layer are listed below.

6.5.1 *Backup& Replication*:

Because of the value of data stored in the Cloud. Organizations can use data recovery and replication methods to avoid data loss. Data is replicated across multiple sites.

6.5.2 *Partitioning*:

Partitioning is a load-control and load-transfer technique used by cloud providers. In multifunctional environments, this balancing and redistribution improve CDBMS productivity and scalability. For cloud data, this layer will use horizontal and vertical partitioning[24].

6.5.3 *Storage*:

Stuff such the data should be stusied on this layer to take less time to access the data. Cloud databases are best served by shared disk architecture. It holds the most current metadata in memory using sophisticated caches.

6.5.4 *Indexing*:

Indexing offers simple data access. For the records that are stored in the database, indexes are established. Cloud applications should index different types of fields to deliver scalable output.

6.5.5 *Load Balancing*:

It must transfer loads throughout servers automatically to ensure that most hardware resources are used efficiently and prevent an overload of resources.

6.5.6 *Fault Tolerance*:

The cloud systems should be configured to cope with failures in the presence of losses to stay operational (perhaps at a lower level). This layer is also responsible for monitoring competitors, recovery strategies, and identifying impasse techniques.

6.5.7 *Security*:

This layer protects the use of encoding/decoding techniques to turn raw data into an unreadable form. These methods may avoid risks such as secrecy, alteration, and manufacturing.
According to[25], Relational databases have implemented relatively safe techniques to provide security services. However, they face numerous security concerns such as SQL injection, cross-site scripting, Root Kits, poor communication protocols.

Many studies are currently being conducted to examine and address these vulnerabilities. In addition, NoSQL databases were created to answer the challenge of storing large amounts of data and increasing database performance. Finally, because the majority of existing applications and software rely on the web, and the amount of data required to store it continues to expand at a rapid rate, we anticipate that NoSQL databases will see significant growth and improvement, as well as overcome their security issues. Because of the value of data stored in the Cloud. Organizations can use data recovery and replication methods to avoid data loss. Data is replicated across multiple sites.

7. Cloud Database Applications

Many cloud database providers provide services that are further classified into three types. A local database program, such as SQL, is used by an effective database, a non-rational database, and a virtual computer. Various companies offer DBaaS, such as Amazon RDS, Google AppEngine Datastore, Microsoft Square Azure, and Amazon SimpleDB as a service[21]. Depending on the standard and form of services offered, each service provider varies from the other. Any criteria can choose the best service the company can provide, which is not exclusive to a particular company. Still, it does aid in determining which service provider is best based on any company's requirements. The following are the most often used databases in cloud computing:

- **PostgreSQL**: Cloud database enables management providers and organizations to have flexible and highly flexible (DBaaS) while releasing DBAs and application designers from current and robust database situations being developed and handled strictly. Postgres Plus Cloud Database reorganizes energy efficiency arrangements when taking advantage of distributed machines. In addition, Cloud Database includes an Oracle-perfect DBaaS, which offers sensational reserve spending funds and adjustments when used with Postgres Plus Advanced Server[26].

- **MongoLab**: MongoDB is an open-source JSON database with a well-organized structure. Dwight Merriman and Geir Magnusson designed it. It is intended to be a good article database instead of a full-quality shop. The data are saved as records of developments of components in JSON. There is versatility in the leading shop and room efficiency. The accommodation is rich, such as records and part of the demand for social databases. The level of versatility is also provided[9]

- **Firebase Real-time Database**: It's a cloud-based NoSQL database that synchronizes data in real-time across all users and allows for offline access. Data is saved as JSON in the real-time database, and all associated clients share one instance, with automated updates to the most recent data[27].

- **Google Datastore**: Google's highly scalable, fully-managed, Google-based NoSQL database service. Cloud storage 'allows users to store data and information in an off-site registry that can be accessed either through the internet or a private network connection' - which is very financially feasible for undertakings since cloud storage archives can replace physical files. Google's Bigtable and Megastore technologies are based on the Cloud Datastore. In Native or Datastore Mode, Google cloud datastore allows users to build databases. Datastore Mode is optimized for smartphone and web applications and new server projects[28].

- **Amazon RDS**: Amazon Relational Database Service (or Amazon RDS) is a distributed relational database service by Amazon Web Services (AWS). It is a web service "in the cloud" that aims at simplifying a relationship database installation, operation, and progression in apps. Servicing processes are managed automatically, such as database patching, backup databases, and time-based restoration. A single API allows scaling storage and measuring resources that call on request for the AWS control aircraft. AWS provides no SSH connection as part of a managed service to the underlying virtual machine[1].

- **Amazon SimpleDB**: Amazon SimpleDB is an Erlang-based distributed database by Amazon.com. It is part of Amazon Web Services and is used as a web service in conjunction with Amazon Elastic Compute Cloud (EC2) and Amazon S3. On December 13, 2007, it was announced[29].

- **Microsoft SQL Azure**: Microsoft Azure SQL Database (previously SQL Azure, SQL Server Data Services, SQL

- Services, and Windows Azure SQL Database) is a managed cloud database (PaaS) offered by Microsoft Azure.

8. Compare Cloud Database Applications

The analysis of the chosen papers is covered in this section. First, relevant articles addressing issues affecting Cloud Databases' functions and performance are found. Then, the first evaluation and comparison presented in table 1 are based on some criteria selected to evaluate the most common cloud database types: SQL and NoSQL databases. Finally, detailed descriptions of the requirements identified to assess functions a performance of Cloud Database are given below.

1. Model: The data model depicts the logical structure of the database. It arranges relevant data and specifies their relationships. Data models are of two sorts Relational and Non-relational[31].

2. Data: Each record may have various attributes or identical attributes.

3. Scale: The scalability of the database is the ability to retain increased data quantities without losing performance. Vertical and horizontal scalability are two forms[31]. Only by incorporating additional hardware resources will you compete, including Memory and Processor, to the current computer in vertical scaling via a single node. On the other hand, each node in horizontal scaling includes only a portion of the data, enabling the existing distributed system community to add more machines[32].

4. Schema: A database schema is a structure that describes how a database is built. It describes the organization of the data and their connection with the relationship between data. Two predefined and dynamic schemes exist—unstructured data in the predefined system. The data structure should be predefined in table form until you start using it. An active schema requires data to be processed before the schema is implemented. Schema depends entirely on how you want data to be processed[31].

Transactions: It supports ACID properties or not. The second evaluation and comparison presented in Table 2 are based on criteria selected to evaluate the most common cloud databases. Detailed descriptions of the requirements that identified to evaluate functions a performance of Cloud Database are given in table 2:

Table 1: comparison between NoSQL and SQL:

	NoSQL	**SQL**
Scale	Horizontal Scale	Vertical Scale
Data	Flexible so that each record may have a different attribute	Same Attribute for every record
Model	Non-relational	Relational
Performance	Max performance if consistency is reduced	Depending on the disk's speed, insert and update performance
Consistency and availability	Utilizes CAP Theorem	High forced consistency prioritized over availability and performance
Schema	Flexible or dynamic	Fixed
Transactions	Dependent on different solutions	ACID transactions
Database types	Key-value, Document Stored, Column Stored, and Graph stored	Table-based database

Table 2: Comparison between Cloud Databases:

No		Google Datastore	Amazon SimpleDB	Mongo Lab	Firebase Realtime Database	Microsoft SQL Azure	PostgreSQL	Amazon RDS(MySQL)
	Type	NoSQL	NoSQL	NoSQL	NoSQL	SQL	SQL	SQL
1	Ease of software portability	Medium/Low	Medium	Medium/High	High	High	High	High
2	Standalone database accessibility	No	Yes	Yes	Yes	Yes	Yes	Yes
3	Transaction capabilities	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4	Max amount data that can be stored	1MB limit on a subset of Data	10GB per database domain	3TB per node	1GB of stored data	50GB per database	16TB per table	1TB per database
5	Configurability and ability to tune database	Low	Low	Medium	High	Medium	High	High
6	Replication	Yes	Yes	Yes	No	Yes	Yes	Yes
7	Designation where the data can be stored	No	No	Yes	Yes	Yes	Yes	Yes
8	cost of using and monthly payment	Starting at no cost, you must pay based on store, read/write, delete, and small operation daily to exceed the free usage.	Pay only for what you use based on Data Transfer, Machine Utilization, and Data Storage.	Pay Monthly based on three plans, including Serverless, Dedicated and Shared instances.	Charges only for bandwidth and storage.	Price calculated based on Single Database or Elastic Pool	free and open-source software	free to try. Pay only for what you use
9	Accessibility.	Using Identity and Access Management (IAM) policies	Using Identity and Access Management (IAM) policies	Access control is enabled	Firebase Authentication	Active Directory authentication (integrated security)	Access Control techniques including Database superusers, Access Privilege, Class removal,	AWS Identity and Access Management (IAM) database authentication

								and schema modification	
10	protocols	MQTT and HTTPS	HTTPS	Wire Protocol	WebSockets to communicate in real-time	TCP/IP	message-based protocol for communication between frontends and backends	SSL/TLS
11	Data backup and recovery	You may create a procedure to periodically backup data from on-premises to Google Cloud using a variety of ways.	Elastic Cloud Gate is a SaaS solution provider to Reduce risk and optimize your disaster recovery response daily, hourly or monthly basis	provides two fully managed methods for backups: Continuous Backups and Cloud Provider Snapshots	users can activate automated backups. This self-service option allows daily backups of your Database application data and rules in JSON format to a Cloud Storage bucket.	Several techniques of backup are available, including (Full/Differential/Log/Copy Only Full) and Compression (Enable/Disable)	several m to backing up data: SQL dump, File system-level backup, and Continuous archiving	Amazon RDS creates and saves automated backups of your DB instance
12	transaction speed	Transactions can last up to 60 seconds, with a 10-second idle time after the first 30 seconds.	stores smaller bits of data and uses less dense drives that are optimized for data access speed	In most cases, transactions improve consistency at the expense of concurrency. As a result, transactions have a significant impact on database performance.	For reading and storing data, Cloud Firestore enables atomic operations. Either all atomic operations succeed, or none of them are applied in a set of atomic actions.	provides unparalleled performance with In-Memory technologies (75,000 transactions per second and reduced query execution time from 15 seconds to 0.26)	When running with a concurrent web server, single row inserts are primarily tied to round trip network latency, with up to 10,000 single row inserts or more on a single node database.	The activity l in your database is measured by database load (DB load). DBLoad is the most critical indicator in Performance Insights, and it is collected every second.
13	index algorithm (which algorithm is supported)	two types of indexes: Built-in indexes and Composite indexes	automatically indexes your data to enable efficient queries	Several index types support: Single Filed, MultiKey, Geospatial, Text and Hashed Index	uses two types of indexes: single-field and composite index	several indexing options including clustered column store, clustered and nonclustered, and heap	several index types: B-tree, hash, GiST, SP-GiST, GIN, and BRIN	spatial index

14	service guarantees	The persistent disk is a Google Cloud Platform high-performance block storage solution that makes it simple to share data and scale without downtime.	A successful write (using PutAttributes, BatchPutAttributes, DeleteAttributes, BatchDeleteAttributes, CreateDomain, or DeleteDomain) guarantees that all copies of the domain will durably persist.	both read concern majority and write concern majority to preserve causal guarantees and durability	The Realtime Database stores key-value pairs in JSON documents synchronized data over WebSockets and works with Android, iOS, and web apps.	Guarantee at least 99.9% availability for load testing operations. We guarantee at least 99.9% availability for development and deployment processes using the premium Azure Pipelines.	load-balancing technique allow several computers to serve the same data	Amazon RDS Multi-AZ deployments give your MySQL databases more availability and durability, making them ideal for production database applications.

9. Rolls of Database Administrator (DBAs) in the Cloud

The job of DBAs changes dramatically depending on whether a business opts for infrastructure-as-a-service (IaaS), database-as-a-service (DBaaS), or a hybrid design[33].

DBA rolls with IaaS.

The DBA is solely responsible for the software running within the virtual machines (VM). The operating system (OS), OS utilities, and database software are included. All of these are chosen and managed by the DBA. On-demand, the DBA can scale up or down a particular VM and build and terminate VMs. DBAs' primary responsibilities do not alter as a result of IaaS. What changes is how DBAs go about doing these things.

Capacity Planning: In an IaaS setting, this planning is quite similar to on-premise in a virtualized environment. The main distinction is that cloud providers will not run out of physical VM hosts, giving them more flexibility. As a result, instead of over-provisioning for a physical machine expected to last three to four years, DBAs should only provide what the business needs today.

Monitoring: Another important activity that DBAs conduct daily is monitoring. Monitoring is available from all of the significant Cloud IaaS providers.

Disaster Recovery and High Availability: High availability (HA) and disaster recovery (DR) designs and tactics are required for mission-critical databases due to their inherent high uptime requirements. To provide effective HA and DR for IaaS, DBAs must understand which clustering technologies are the most cloud-friendly, such as whether a share-it-all cluster or a local storage cluster is preferable or easier.

DBA Roles with DBaaS.

The DBA is relieved of a load of managing infrastructure, operating systems, and RDBMS patching and maintenance using DBaaS. Furthermore, high availability and data security are built-in features. There is no need to over-provision in anticipation of future expansion because scalability is elastic. Because DBaaS intends to automate most, if not all, operational operations, the essential tasks of a DBA do change with DBaaS. (see Table 3).

Table 3: DBA role with IaaS and DBaaS:

DBA role with IaaS	DBA role with DBaaS
Manage the OS	No access to the OS
Install patches No patching	No patching
Upgrade the system One-button upgrade	One-button upgrade
Perform backup and recovery	Backup and recovery performed by a service provider
Determine and implement HA and DR architecture and strategies	HA and DR are inherently built-in

Instead of infrastructure maintenance and assistance, the DBA position can now be refined to operational activities. Schema evolution, data management, data ingestion, and data recovery due to operating loss, such as the accidental dropping, require DBA abilities. The DBA can now concentrate on query performance and accomplishing business goals rather than platform maintenance.

Tuning the performance: with DBaaS, performance tweaking changes as well. A DBA must be familiar with the mechanisms provided by the service provider for increasing or decreasing compute power and the knowledge to select the most effective and cost-effective solution. The DBA can alter the Google Cloud SQL deployment instance type in the Google Cloud Platform. The DBA can also change the database transaction units or adopt a core-based sizing approach in Microsoft Azure[33].

10. Discussions

This study aims to discover prior research efforts on Cloud database concerns. Looking at the comparison of table 1, SQL databases are vertically scalable. This raises the expense of vertical scaling. Another drawback to vertical scaling is based on a single machine; therefore, the program often falls as the server falls. SQL Databases can also be run on a distributed system. However, NoSQL databases can be scaled horizontally, which reduces the expense and speed of horizontal scaling. For unstructured data, SQL requires a predefined schema. Before using SQL to manipulate data, you must first create a table-based data structure. However, no predefined schema is mandatory for a NoSQL database. For unstructured files, NoSQL uses a dynamic scheme. So, if you want real-time data – NoSQL needs no schemes so that the data process is faster than SQL. SQL databases offer ACID, which guarantees data consistency and integrity. In addition, SQL is a better fit for complicated query contexts than NoSQL when working with complex queries and reports.

Most modern NoSQL databases scale horizontally, which means you can expand your database by adding more servers. Increased load is handled via sharding or adding additional servers to your NoSQL database. In most NoSQL databases, the BASE consistency model provides" Eventual Consistency" to maintain high scalability and availability. However, the ACID consistency model focuses on "Strong consistency or write consistency" in SQL databases. NoSQL is substantially quicker than relational databases in reading and writing speed, especially in key-value storage systems. In use instances such as online transactions, this means decreased waiting time.

On the other hand, If we look at the comparison of table 2, It is clear that the DBaaS is the best and has more options and better configuration, performance, and features are Amazon RDS. However, a smaller company or business would use a cheaper option, like Amazon SimpleDB. There is much more to go through in the sense of challenges before a user. Still, this paper only reviews the different DBaaS services and which one has advantages over the other and can be better suited for one's business/company; our pick would be Amazon RDS as it is the better option among the rest of them.

11. Conclusion

The cloud database framework was presented in this study, and some of its key features were examined. Businesses have grown to rely on cloud computing for a variety of reasons. Cloud computing services have already begun to emerge as a way to boost and speed up data sharing without the need for each firm to set up its own data center. The cloud database has recently formed a new (DBaaS), allowing enterprises and agencies to use DBaaS suppliers' resources and invest in and maintain data centers' hardware and apps holding all of the database's information without any issues. A comprehensive evaluation presented and based on the comparisons, NoSQL databases have proven better for most jobs. They are reliable and perform better and faster. Also, Amazon RDS is the best and has more options and better configuration, performance, and features; we recommend it for almost all organizations. In addition, for personal use or group, Firebase Realtime Database is ideal, especially if you do not have 24/7 access to the Internet and want to have a specified storage location.

From the reviewed papers and our previous sections, we can give some recommendations which concluded as follow:

- NoSQL databases have proven better for most jobs, extensive corporation facilities, and organizations; they are fast, reliable, and perform better and faster.
- Amazon RDS is the best and has more options and better configuration, performance, and features; we recommend it for almost all organizations.
- Small businesses should use Amazon SimpleDB as it is cheaper and better suited for smaller-scale data.
- Firebase Real-time Database is ideal for personal use or group, especially if you do not have 24/7 access to the Internet and want a specified storage location.

Conflict of interests

None.

References

1. G. Narasimhan. (2018) *Database Migration on Premises to AWS RDS.* EAI Endorsed Transactions on Cloud Systems. From: https://eudl.eu/doi/10.4108/eai.11-4-2018.154463
2. N. Nix. CIA tech official calls Amazon cloud project 'transformational'. Bloomberg, June 2018. https://www.bloombergquint.com/politics/cia-tech-official-calls-amazon-cloud-project-transformational. Last accessed 14/12/2021
3. A. Verbitski et al. Amazon Aurora: Design considerations for high throughput cloud-native relational databases. In Proc. ACM Int. Conf. on Management of Data, SIGMOD '17, pages 1041–1052, (2017).
4. M. Babacar, M. Ana, Q.Gustavo, M.Rita, C.Daniela (2021)."A Domain-Specific Language to Provide Middleware for Interoperability among SaaS and DaaS/DBaaS through a Metamodel Approach". From: https://www.scitepress.org/Papers/2021/104596/104596.pdf
5. F. Elivaldo et al. Transparent Interoperability Middleware between Data and Service

   Cloud Layers. (2018). International Conference on Cloud Computing and Services Science

   From: https://www.scitepress.org/Papers/2018/67041/67041.pdf
6. Antonopoulos, P., et al. (2019, June). Socrates: The new SQL server in the Cloud. In Proceedings of the 2019 International Conference on Management of Data (pp. 1743-1756).
7. M. Steffen et al."Benchmarking the Performance Impact of Transport Layer Security in Cloud Database Systems". (2014). IEEE International Conference on Cloud Engineering (IC2E).
8. A. Peshraw, V. Asaf. " Advantages to Disadvantages of Cloud Computing for Small-Sized Business" (July 2019). 7th International Symposium on Digital Forensics and Security (ISDFS).IEEE
9. A. Manar and I. Abdellah. Database-as-a-Service for Big Data: A Review. (2017). International Journal of Mathematical and Computational Methods. From: https://www.iaras.org/iaras/filedownloads/ijmcm/2017/001-0027(2017).pdf
10. M. Ayesha, M. Muhammad. "Security Framework for Cloud Computing Environment: A Review" Journal of Emerging Trends in Computing and Information Sciences. March 2012
11. B. Harrison and Babak. "Databases in Cloud Computing: A Literature Review". I.J. Information Technology and Computer Science, 2017, 4, 9-17
12. F. Azhi, R. Bilal, and S. Twana. Comparative Study of Relational and NON-Relations database performances Using Oracle and MongoDB Systems. Volume 5, Issue 11, November (2014), pp. 11-22. IAME: www.iaeme.com/IJCET.asp.
13. Wang, R., & Yang, Z. SQL vs NoSQL: A Performance Comparison. (2017). From:

    https://www.cs.rochester.edu/courses/261/fall2017/termpaper/submissions/06/Paper.pdf
14. M. Christine. What is Cloud? It is Time to Update the NIST Definition. (Jun. 2018), pp. 6-11, vol. 5. IEEE Cloud Computing
15. M. Mohammad, Z. Rafiqul. A Systematic Review on Cloud Computing. Vol.-6, Issue-11, Nov 2018.
16. O. Isaac, M. Ananya, A. Frank, and G. Rowland: Cloud Computing Architecture: A Critical Analysis. August 2018. 18th International Conference on Computational Science and Applications (ICCSA)
17. R. Aaqib and C. Amit. Cloud Computing Characteristics and Services: A Brief Review.

International Journal of Computer Sciences and Engineering (IJCSE)Vol.-7, Issue-2, Feb 2019.

18. Sharma, S., Singh, P., & Singh, A. (2016). User-centric security requirements and threat analysis in Cloud Computing. Computing, 2(04).

19. Curino, C., Madden, S., and et.al. Relational Cloud: A Database-as-a-Service for the Cloud. Retrieved November 24 2012, from: https://people.csail.mit.edu/nickolai/papers/curino-relcloud.pdf

20. Xi Zheng. (2018). Database as a Service - Current Issues and Its Future

    From: https://arxiv.org/pdf/1804.00465.pdf

21. A. Adeel, M. Muhammad, and A. Waleed.Cloud based Network Management and Control for Building Automation. (February 2017). IEEE, International Multi-Topic Conference (INMIC).

22. S. Sakr, A.Lui, Daniel M. Batista, and M. Alomari: "A Survey of Large Scale Data Management Approaches in Cloud Environment". IEEE Communication Surveys and Tutorials, Volume 13(3), 2011, page 311-336

23. Bashir Alam et al. 5-Layered Architecture of Cloud Database Management System. AASRI Procedia 5 (2013) 194 – 199

24. Yvette E. Gelogo and S. Lee, "Database Management System as a Cloud Service", International Journal of Future Generation Communication and Networking, Vol. 5(2), 2012, pages 71-76.

25. C. Warner and P. Chris. The Evolution Role of DBAs in The Cloud. (2021).

26. F.M. Dahunsi, A. J. Joseph , O. A. Sarumi , O. O. Obe. Database Management System for Mobile Crowdsourcing Applications. Nigerian Journal of Technology (NIJOTECH) Vol. 40, No. 4 July, (2021), pp.713 –727

27. Laurence Moroney " The Definitive Guide to Firebase 2017 (pp.51-71)"

28. "Google Cloud Datastore". Quickstart. Retrieved April 7, 2021.

29. O. Michael, O. Adakole, and O. Ayomide. Cloud Systems and Applications : A Review.

    Volume 7, Issue 1 Page Number: 142-149 (February 2021).International Journal of Scientific Research in Computer Science, Engineering and Information Technology

30. Byham, Rick (August 30, 2016). "Azure SQL Database Transact-SQL differences".

31. Pore, S. S., & Pawar, S. B. (2015). Comparative Study of SQL & NoSQL Databases. International Journal of Advanced Research in Computer Engineering & Technology, 4(5), 1747-17

32. Li, Y., & Manoharan, S. (2013, August). A performance comparison of SQL and NoSQL databases. In 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) (pp. 15-19). IEEE.

33. M. Mohamed, A. Obay, and I. Mohammed. Relational vs. NoSQL Databases: A Survey. International Journal of Computer and Information Technology. Volume 03 – Issue 03, May 2014.