|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Model** | **Scalability** | **Flexibility** | **Performance** | **Complexity** | **Structure** |
| **Key-Value Store** | High | High | High | None | Primary key with some value |
| **Document-Oriented Store** | Variable | High | High | Low | Tree structured JSON Object Form |
| **Column-Based Store** | High | Moderate | High | Low | Rows and Columns |
| **Graph-Based Store** | Variable | High | Variable | High | Graph entities and relations |

|  |  |  |
| --- | --- | --- |
| **Parameter** | **RDBMS** | **NoSQL** |
| **Software Distribution** | Both Open-source and Closed-source | Mostly Open source |
| **Cost** | Expensive | Cheaper |
| **Volume of Data** | Can Limited Data | Handles large volumes of data |
| **Scalability** | By upgrading hardware of a single server | Horizontal scaling using commodity servers |
| **Complexity** | Complex data is difficult to convert into tables | Relatively less difficult to store complex data |
| **Consistency** | Strong since uses strict schemas design | Poor since uses schema less design |
| **Performance** | Slow to process information | Better performance to process information |
| **Data Manipulation** | Most RDBMS only use SQL as the query language | NoSQL databases have their own programmatic APIs for querying |
| **Security** | Strong mechanisms to protect the data | No built-in security protocols but can be handled by middleware programs. |

There are various parameters to determine the better type of database over other. Some of the most important features and criteria for comparison of the database management system are

Software Distribution

This may be one of the key factors that can the decide the database for an application. Relational databases are available in both open-source and closed-source whereas most of the NoSQL database systems are open-source programs. Proprietary database systems tend to have greater security but lack personalization.

Cost

Price of the database systems can be a deciding factor if the application, the database is used for, needs large amounts of data storage which may lead to great amount of investments. Most proprietary database systems are expensive when compared to open-source database systems. Additional hardware upgrades also add up to the database maintenance charges. While most NoSQL databases are open-sourced, they tend to be inexpensive and can also be modified to personal use. By using virtual machines instead of commodity servers and leveraging the benefits of inexpensive software, one can further reduce the investments.

Volume of Data

The emergence of internet and social media applications on web have increased the volume and variety of data. Thus, the databases are expected to handle such large amounts of data effectively and efficiently. Relational databases have failed to handle big data because it lacks in high velocity since it was designed more towards steady data processing rather than rapid growth. This inability of RDBMS led to large tech companies like Google, Facebook shift towards NoSQL which has proven to handle data intensive applications.

Scalability

Both RDBMS and NoSQL databases are usually scalable, but RDBMS have a certain limit and this limit is particularly hardware dependent which are sometimes fixed by design and unaltered. Hardware upgrading a server, or a database increases the effort for the database administrator. Whereas NoSQL offers great scalability by horizontal scaling and is simple, cheaper, and not restricted by hardware limitations. This is one of the reasons why large tech and social media companies are migrating to NoSQL.

Complexity

Relational databases strictly follow database schemas for creating databases and tables which makes it difficult to convert raw data into relation tables. Also, relational databases require large and complex queries to read, modify or create data. Whereas NoSQL required no design schemas or relations between the data or no large and complex queries read, modify, or create data. NoSQL also supports both structured and unstructured data which makes it easier to store any form of raw data. Therefore, no skill is comparatively less complex and simple than relational databases.

Consistency

Since relational database tables our design using strict schemas, they offer strong consistency. This provides uniform you of data but with less flexibility and availability. This makes sense since relational databases are required to follow ACID properties which strictly promotes consistency in the database systems. NoSQL databases are designed to support wide variety of data and since the data is created without no strict schemas, they offer less consistency. NoSQL aims to provide more flexibility and availability rather than consistency.

Performance

RDBMS require much time to load and process the data then NoSQL database system. relational databases serve information primarily from non-volatile memory whereas NoSQL systems serve the data from volatile memory. Thus, by design time taken to retrieve the information from RDBMS is comparatively longer than the time taken by NoSQL databases.

This is one of the main reasons for large web applications to use NoSQL databases rather than relational databases.

Data Manipulation Language

Relational databases are confined to use a single query language for the data manipulation. Most of the relational databases use SQL for this purpose And SQL has become a standard with great documentation and wide community . However, some databases have their own variation of SQL language. Since there is only one universal language which works with most of the relational databases, developers do not have much problem using a new relational database.

NoSQL databases does not have a standard of any kind for querying by design and does not have a query language. NoSQL databases provides an API which can be used to query data. So, these databases support programmatic querying through their own unique APIs.

Some NoSQL databases like MongoDB support various popular programming languages like Python, JavaScript, Scala for data querying directly from their applications.

Security

Database systems, in general, face some serious security challenges such as SQL/NoSQL injection and cross site scripting. Security is a substantial part of relational database systems where are in the NoSQL database systems security is handled by middleware programs. So, in RDBMS there's too little effort needed to secure a database from potential threats. Whereas in NoSQL database systems the developer or the administrator must need to put in some effort to configure and setup the required security for the applications. Also, these security measures must be implemented in such a way that it does not interrupt or compromise the databases’ flexibility, scalability, and performance.