SQL was first developed at IBM. With SQL, you can easily query a database to add, alter, update, or delete data in a plain English-like language. You can use SQL to manage multiple relational database management systems, such as MySQL, SQL Server, Oracle, and MS Access. By writing codes and queries in SQL, you can create and modify any database along with the creation of schemas. Whereas NoSQL databases are primarily used in big data and real-time web applications. These types of databases gain a surge in their popularity in the early twenty-first century. A primary reason for this sudden surge can say to be triggered by companies such as Facebook, Amazon, and Google.

**Differences between SQL and NoSQL**

**The Structure**

SQL databases are table-based on the other hand NoSQL databases are key-value pairs, document-based, graph databases or wide-column stores. This makes relational SQL databases a better option for applications that require multi-row transactions such as an accounting system or for legacy systems that were built for a relational structure.

**The Support**

Great support is available for all SQL database from their vendors. Also a lot of independent consultations are there who can help you with SQL database for a very large scale deployments but for some NoSQL database you still have to rely on community support and only limited outside experts are available for setting up and deploying your large scale NoSQL deployments.

Some examples of SQL databases include PostgreSQL, MySQL, Oracle and Microsoft SQL Server. NoSQL database examples include Redis, RavenDB Cassandra, MongoDB, BigTable, HBase, Neo4j and CouchDB.

* **The Property followed:**

SQL databases follow ACID properties (Atomicity, Consistency, Isolation and Durability) whereas the NoSQL database follows the Brewers CAP theorem (Consistency, Availability and Partition tolerance).

|  |  |
| --- | --- |
| SQL | NoSQL |
| Relational Database Management System (RDBMS) | Non-relational or distributed database system |
| These databases have fixed or static or predefined schema | They have dynamic schema |
| These databases are not suited for hierarchical data storage. | These databases are best suited for hierarchical data storage. |
| These databases are best suited for complex queries | These databases are not so good for complex queries |
| Vertically Scalable | Horizontally scalable |
| Follows ACID property | Follows CAP(consistency, availability, partition tolerance) |