SQL and NoSQL Databases

Meenu Dave
Jagan Nath University, Jaipur, India

Volume 2, Issue 8, August 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

SQL and NoSQL Databases

Vatika Sharma[1], **Meenu Dave**[2]

[1]*M.Tech. Scholar, Department of CSE, Jagan Nath University, Jaipur, India*
vatika.sharma15@gmail.com
[2]*Assistant Professor, Department of CSE, Jagan Nath University, Jaipur, India*
meenu.dave@jagannathuniversity.org

Abstract: NoSQL (Not only SQL) is a database used to store large amounts of data. NoSQL databases are distributed, non-relational, open source and are horizontally scalable (in linear way). NoSQL does not follow property of ACID as we follow in SQL. In this research paper, we are surveying about NoSQL, its background, fundamentals like ACID, BASE and CAP theorem. Also on the basis of CAP theorem, study is carried out about the various types of NoSQL data stores with their examples, characteristics, and pros and cons of NoSQL.

Keywords: ACID, BASE, SQL, NoSQL, CAP, CURD.

I. INTRODUCTION

NoSQL stands for Not Only SQL. It is pronounced as **noseequel**. It is one of the another type of data storage other than databases (that were used earlier) that is used to store huge amount of data storage like data in facebook (which keeps on increasing day by day). NoSQL is a non-relational database management system (sometimes called as derived from relational database), fast information retrieval database and is portable. NoSQL basically derives from RDB database system. This database usually interacts with the UNIX operating system. NoSQL databases are those databases that are non-relational, open source, distributed in nature as well as it is having high performance in a linear way that is horizontally scalable. Non-relational database does not organize its data in related tables (i.e., data is stored in a non-normalized way). NoSQL databases are open source; therefore, everyone can look into its code freely, update it according to his needs and compile it. Distributed means data is spread to different machines and is managed by different machines so here it uses the concept of data replication. NoSQL may be symbolically represented as shown in figure 1:

Figure 1: Symbolic representation of NoSQL

Figure 1 states the querying to database without any interaction or interface of SQL language. The tilted line in the figure shows database usage without using SQL (Structured Query Language). So, to access these databases we can use some other formats like XML to store and retrieve information from the database.

With the advent of social networking sites like facebook and twitter, the demand of new technology that can handle huge amounts of data has lead the emergence of various new technologies and one of the prominent is NoSQL which is quite helpful in data warehousing. NoSQL (non-relational) is comparatively faster than relational databases.. Previously, in SQL, we were using Query language to fetch as well as to store data; for NoSQL we store large data entities using documents in XML (eXtensible Mark up Language) formats. XML language is basically used to store structured data in a human readable form. This is further discussed in later sections.

II. BACKGROUND

In 1960's Multivalue databases were developed so as to perform some hierarchal database operations using B+ tree storage. According to Scott Jones M[umps] was developed in 1966 to act upon such types of databases (hierarchal). In 1977, M[umps] was approved to be an ANSI standard language. In 1979, Ken Thompson developed DBM (Database Manager) which uses hashing techniques to provide user the faster retrieval of data. In 1980's large number of successors to DBM were developed which are as follows:

- TDBM is a DBM type database which supports atomic transactions (that follow ACID properties A: Atomicity, C: Consistency, I: Isolation, D: Durability). The advantages of TDBM include:

  (1) If the application is using TDBM, in case of damages or crashes, the contents stored in the database will be safe.
  (2) TDBM also supports nested atomic transactions using multithreading operations.
  SDBM is a DBM type database used for some licensing reasons.
- GT.M is another successor to DBM that focuses on transaction processing having high performance. In 2000, this DBM was open sourced [4].

Evolution of the SQL databases begun in the late 1990s. After a few years it became a serious competitor to RDBMS (Relational Database Management Systems) [4]. In the year 2009 and 2010 there were organized NoSQL conferences like NoSQL live, NoSQL eu, NoSQL east, etc. This NoSQL name was first used by Carlo Strozzi in 1998 as the name of the file he was developing for his database.

Now a days, NoSQL is becoming so popular due to its high storage and also because its properties avoid the basic features of SQL.

Relational databases are designed to run on a single machine, so for this we need a large machine to scale. One solution to overcome this is to buy lots of machines (and turn them into a cluster) to store data. It will be a cheaper process as well as horizontally scalable (linear way) also. Even if a machine is down, the overall reliability of the cluster is quite high. Thus the dominance of SQL is declining and NoSQL is gaining popularity.

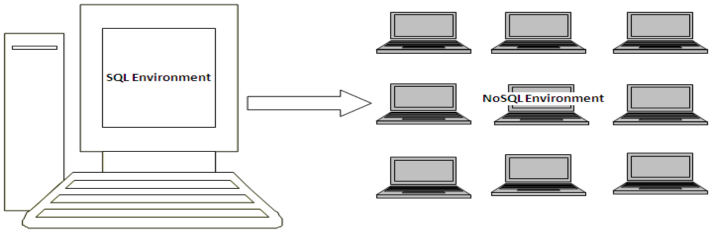


Figure 2: Decline in dominance of SQL

III. AXIOMATICS OF NoSQL

A. ACID free

ACID stands for Atomicity, Consistency, Isolation and Durability. ACID concept basically comes from the SQL environment. But in NoSQL we will not use the ACID concept because of Consistency feature of SQL [6]. In this paper we will see how ACID concept creates problems to NoSQL.

As in the distributed environment, data is spread to different machines, each machine stores its data and maintenance of consistency is needed. For example, if there is change in one tupple of the table then changes are needed in each and every machine on which that particular data resides. If information regarding an updation spreads immediately, then consistency is given; if not, then inconsistency is carried out.

B. BASE

BASE stands for Basically, Available, Soft state, and Eventual consistency. BASE is reverse of ACID [1]. NoSQL databases are divided in between the road from ACID to BASE. After a transaction consistency the state that we will get is soft state not a solid state. The main focus leading behind the BASE is the permanent availability [6].

For example, thinking about the databases in banks, if two persons are accessing the same account in different cities then data updations is needed not just in time but needs some real time databases as well. Those updations need to be done frequently on all machines. Some more examples are online railway reservation, online book trade, etc.

C. CAP

CAP stands for Consistency, Availability and Partition tolerance. CAP is basically a theorem that follows three principles [1][2]:

(1) The data available on all machines should be same in all respects and updations to be made on all machines frequently i.e. consistent data.
(2) Data must be available permanently and should be accessible each and every time i.e. availability.
(3) During machine failure or any faults in the machines database going to work fine without stopping their work i.e. partition tolerance.

IV. ROLE OF DATA ARCHITECTURE IN NOSQL

A. **Components:** There are four components in its building block [6].

[1.] **Modelling Language:** It describes the structure of the database and also defines schema on which it is based. Data is stored in the form of rows and columns using XML formats. And each data (value) corresponding to it is assigned a key that is unique in nature. For faster access of data, the model is built in a suitable environment.
[2.] **Database Structure:** Each and every database while building uses its own data structures and stores data using permanent storage device.
[3.] **Database Query language:** All the operations are performed on the database that are creation, updation, read and delete (CURD).
[4.] **Transactions:** During any transaction in the data, there may be any type of faults or a failure; then, the machine will not stop working.

B. NoSQL data store types

On the basis of CAP theorem NoSQL databases are divided into number of databases. There are four new different types of data stores in NoSQL [4].

1. **Key value databases:** The key value databases name itself states that it is a combination of two things that is key and a value. It is one of the low profile (traditional) database systems. Key Value (KV) databases are mother of all the databases of NoSQL.

- Key is a unique identifier to a particular data entry. Key should not be repeated if one used that it is not duplicate in nature.
- Value is a kind of data that is pointed by a key.

Key value databases seem to be as hash tables or look up tables. In this type of database, there is only one way to query that is with the help of key (unique) and all the keys may name in any data objects and are arranged in an alphabetical order [3]. For higher availability of data stores data objects are replicated.
This is as illustrated in figure 4 below. For example, let's take an example of bank database as shown in figure 3.

BANK DATABASE	
Key	Value
1	ID:1 Joining Date: 15-July-1985 Designation: Cashier
2	ID:2 Joining Date: 19-March-1982 Designation: Manager
3	ID:3 Joining Date: 4-April-1988 Designation: Front Desk Officer

Figure 3: Key Value (KV) databases

In the given figure there are two columns representing key and a value. Here key is unique and representing their values or attributes corresponding to it and data is represented in the form of ring and the partitioning of data is done on the basis of their alphabets (in sorted order) and data is also replicated in the form of ring. This is as represented in next section.

1.1 Characteristics of Key value databases

- Number of keys can have a dynamic set of attributes in the key value databases during storage of data.
- Data stored in the database is stored in the alphabetical order.

- All the activities can be performed on the data i.e. CRUD (Create, Read, and Update and Delete).
- All the relationships to the data are stored in the application code (not explicitly spread).

1.2 Key Value (KV) databases uses

- It is one of the simple data model among all (we will discuss later) as it uses only key and a value. (as depicted in figure 3)
- It handles huge data load.
- It scales to large volume of data.
- Replication of data is done using database in the form of ring. The replicated data is stored in the form of ring as well as in the alphabetical order. This is as shown in figure 4 below:

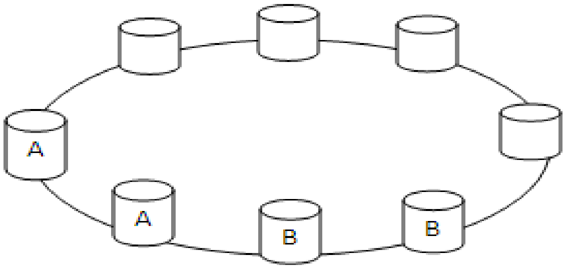


Figure 4: Ring Partitioning and replication of data

2. Document Stores Databases

Document Stores databases are those NoSQL databases which use records as documents. This type of database store unstructured (text) or semi-structured (XML) documents which are usually hierarchal in nature. Here each document consists of a set of keys and values which are almost same as there in the Key Value databases. Each database residing in the document stores points to its fields using pointers as it uses the technique of hashing. Document Stores Databases are schema free and are not fixed in nature.

The structure of Document Stores Databases is as illustrated in figure 5 given below. The figure depicts that it consists of number of databases in the document store such as databases 1,2,3,4 and is having its id A, B, C, D residing in it which is pointing to its database that is having some relation to it. Databases point to its value using some unique key residing in its database. This consists of an array of databases (that is in form of buckets).This will be clearer after taking an example discussed below.

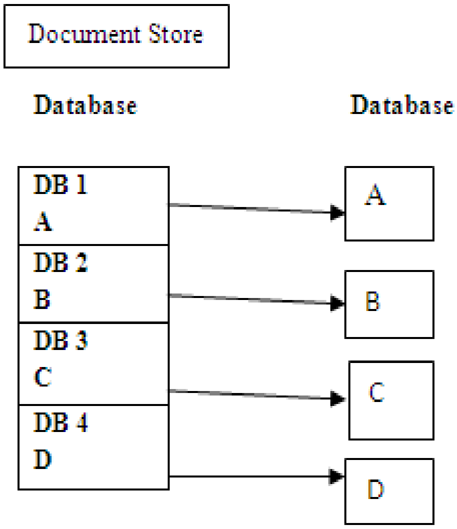


Figure 5: Structure of document stores database

2.1 Characteristics of Document Stores Database

- Documents are addressed in the database using key (unique) that represents that document.
- There are number of varieties to organize data that is collections, tags, non-visible metadata and directory hierarchies.
- In this we can use a key-value lookup to retrieve a document.

2.2 Example of Document Store databases

Figure 6 illustrates an example of Document Store Databases. In this figure we are taking an example of bank database which uses ID as a unique key. And it also comprises of ID database which contains all the information related to the person who is having that particular ID. An ID in the bank database points to the ID database using pointers.
The bank database comprises of two documents:
(1) A person having first name: Bob; Last name: Jackson; Address: Z Street, England; ID: A100.
(2) A person having first name: sherry; Last name: Hilton; Address: Denver, USA; ID: A101.

Now we can consider ID of both documents as a key that is used for correspondence. Key should be unique in nature and is not null. We cannot create a duplicate key. We selected ID as a key because name, address cannot be unique and ID of a person is always unique. So, that's why we have chosen ID as a key. Now we are having one another database that is ID database. The ID database comprises of person's salary and designation. ID database comprises of databases of ID's A100, A101. Documents existing in a bank databases comprises of ID's and points to its particular ID's data. This is how data is organized in a document store databases. Data is organized in the form of collections, tags, metadata, etc. In this we can retrieve data using pair of key and value. This is better illustrated in the example where we have key A100 and its corresponding attributes consists of its salary with designation.

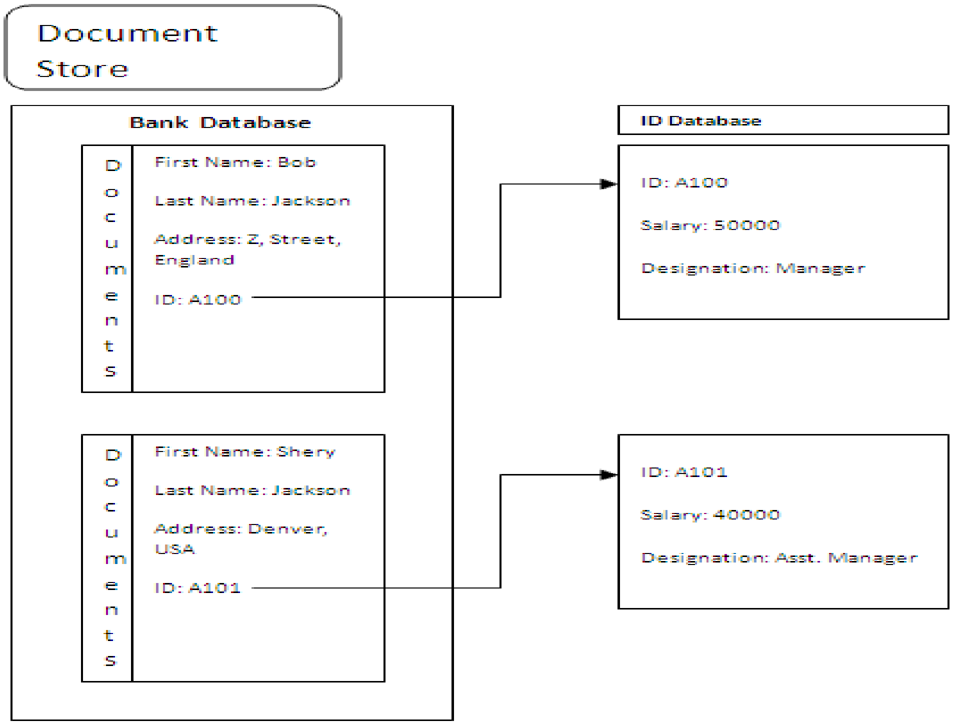


Figure 6: Document Store Databases

3. Columnar Databases

Columnar Databases are also known as column family databases because they are column-oriented databases.

There are two types of column oriented databases whose detail is as given below:

(1) Wide-Column data stores:

It is one type of NoSQL database. Wide Column data stores are those databases that are used for processing of web, streaming of data and documents. The structure of wide Column data store is as depicted in figure 7 below:

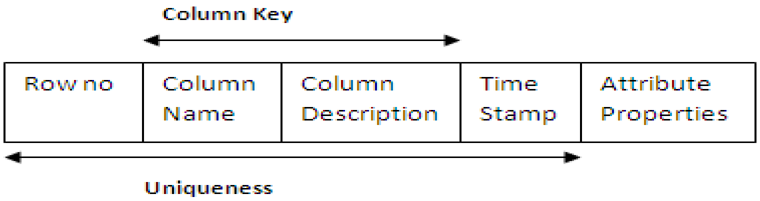


Figure 7: Structure of Wide column data stores

Meaning of each field provided in the structure of wide column data store databases is as depicted below in Figure 7.

ATTRIBUTE	MEANING
Row no	It is a key that is unique in nature. It may be a string or a number.
Column Name	Data stored on the basis of column family.
Column Description	It describes the stored data item.
Time stamp	It tells the complete time of particular instance.
Data value	Value or attributes related to that corresponding key.

Figure 8: Meaning of fields in structure of wide column databases

(2) Column oriented databases:

To understand column oriented databases let's take an example of bank database given in figure 9 whose attribute fields are EmpID, Salary and designation and values corresponding to it are as depicted in database.

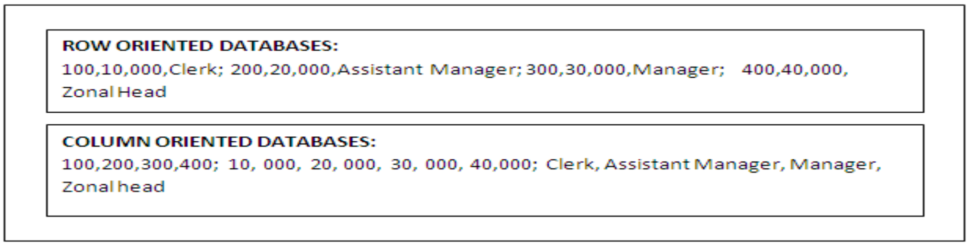
EmpID	Salary	Designation
100	10,000	Clerk
200	20,000	Assistant Manager
300	30,000	Manager
400	40,000	Zonal Head

Figure 9: Example of Bank database

Representation of Row oriented databases and column oriented databases:

- Row oriented databases are those databases in which all the rows are put together one by one.
- Column oriented databases those databases in which all the values containing columns are put together.

With the help of the database given above we will represent row and column oriented databases which is as shown in below figure 10:



ROW ORIENTED DATABASES:
100,10,000,Clerk; 200,20,000,Assistant Manager; 300,30,000,Manager;   400,40,000, Zonal Head

COLUMN ORIENTED DATABASES:
100,200,300,400; 10, 000, 20, 000, 30, 000, 40,000; Clerk, Assistant Manager, Manager, Zonal head

Figure 10: Representation of row and column oriented databases

3.1 Characteristics of columnar databases

(1) Columnar databases are faster than row based databases while querying.
(2) In columnar databases, assignment of storage unit is done to each and every column.
(3) In the columnar DBMS only the required columns are read, so reading is faster in this case.

4. Graph databases

Graph databases are based on the graph theory. In general, we see that graph usually consists of \nodes, properties and edges. NoSQL Graph database consists of:
(1)Nodes represent entities
(2)Properties represent attributes
(3)Edges represent relationships [6].

The structure of graph database is as shown below:

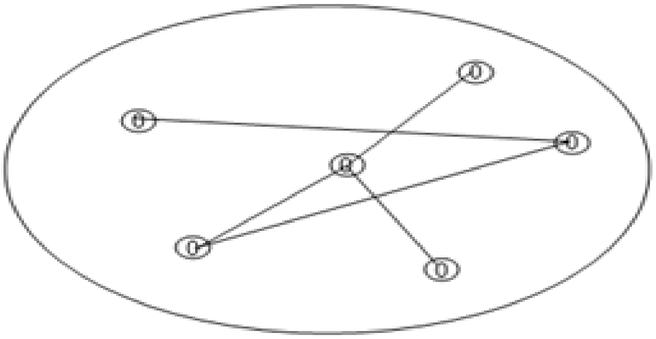


Figure 11: Structure of graph database

4.1 Example of Graph databases

Example of Structure of Graph database is shown in Figure 12. It explains us how to represent nodes, relationships and properties. Here there are two nodes (represented by circles) A and B which are representing BANK and ACCOUNTS. Both are having a relationship represented by a line where a bank has number of accounts. Next is the properties/attributes/values which represents types of accounts that is in the example given in the following figure. Properties are Savings, Current and RD.

4.2 Characteristics of Graph databases

- Graph traversals are executed with constant speed independent of total size of the graph. There are no set operations involved that decrease performance as seen with join operations in RDBMS.
- Graph databases are having high performance in context to their deep traversals.
- These are used for shortest path calculations.
- These are scalable. But its complexity increases.

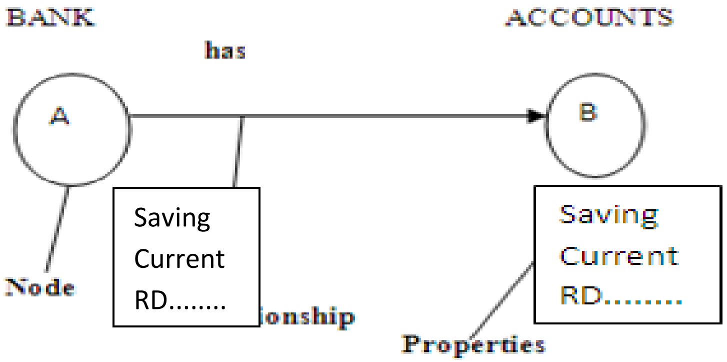


Figure 12: Example of structure of Graph databases

V. CHARACTERISTICS OF NoSQL

- NoSQL does not use the relational data model thus does not use SQL language.
- NoSQL stores large volume of data.
- In distributed environment (spread data to different machines), we use NoSQL without any inconsistency.
- If any faults or failures exist in any machine, then in this there will be no discontinuation of any work.
- NoSQL is open source database, i.e. its source code is available to everyone and is free to use it without any overheads.
- NoSQL allows data to store in any record that is it is not having any fixed schema.
- NoSQL does not use concept of ACID properties.
- NoSQL is horizontally scalable leading to high performance in a linear way.
- It is having more flexible structure.

VI. CONCLUSION AND FUTURE WORK

The main aim of this paper is to give an overview of NoSQL databases, about how it has declined the dominance of SQL, with its background and characteristics. It also describes its fundamentals that form the base of the NoSQL databases like ACID, BASE and CAP theorem. ACID property is not used in the NoSQL databases databases because of data consistency so we get to know how SQL lags data consistency. Later, on the basis of the CAP theorem we described different types of NoSQL databases that are Key-Value databases, Document Store Databases, Columnar based databases and Graph databases with the help of an examples. In addition to all these we have also described their characteristics, complexity and performance. Further research is going on in the new technologies that are arising for or after NoSQL that is polygon persistence, etc.

REFERENCES

[1.] SilvanWeber, " NoSQLDatabases " http://www.christof-strauch.de /nosqldbs.pdf

[2.] Martin Fowler and Pramod Sadalage Rendered, "NoSQLdbs- ", February8,2012,11:26, http://martinfowler.com/articles/nosql-intro.pdf

[3.] An Oracle White Paper, "Oracle NoSQL Database", September2011, http://www.oracle.com/technetwork/database/nosqldb/learnmore/nosql-database-498041.pdf

[4.] Luis Ferreira Universidade do Minho, "Bridging the gap between SQL and NoSQL", httpsikhote.files.wordpress.com201105artigo-mi-star1.pdf

[5.] Andrew J. Brust, Blue Badge Insights, Inc., "NoSQL and the Windows Azure platform", April 25, 2011

[6.] DAMA - Philadelphia / Delaware Valley, the "Role of Data Architecture in NOSQL", Wednesday January 11th, 2012, http://www.damaphila.org/HaugheyNOSQL.pdf