

A Survey and Comparison of Relational and Non-Relational Database

Nishtha Jatana¹Sahil Puri²Mehak Ahuja³Ishita Kathuria⁴Dishant Gosain⁵¹Assistant professor, Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, New Delhi, India^{2,3,4,5} Student, Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, New Delhi, India

Abstract

We report the comparison between the two leading type of Database storage components prevailing in the industry. The Database is largely concerned with managing massive amount of data in a consistent, stable, repeatable and quick manner. The prominent features of both relational as well as non relational databases have been specified which form the basis of the comparison between the two types of database. The relational model is based on mathematical theory(set theory, relational theory) whereas the non-relational databases may or may not have a single groundwork mathematical theory. Relational model is beneficial when it comes to reliability, flexibility, robustness, scalability requirements but in order to cater to the needs of modern applications where the data is huge and generally unstructured; Non-relational databases show true signs of usability here. Based on the characteristics, commonly used tools of relational and non relational databases are mentioned along with brief introduction of the tools. Comparison has been done between the tools to notify the distinctive features of tools of relational and non relational database. Conclusive remarks about the two categories of database have been mentioned.

1. INTRODUCTION

Our research comprises of studying various relational and non-relational databases for storing and retrieval of large amount of data. The attributes and features of both are listed in the following columns.

Database is a collection of large volumes of data. Database Management System is an organized way of managing, retrieving and storing the data. The few actions that can be performed on database over any other storing mechanism are as follows:

- Perform complex calculations
- Retrieve records on the basis of matching or comparing functions
- Update bulk records at an instance.

- Information within multiple tables can be co-related.

The reliability of database model is checked with the help of ACID properties.

- *Atomicity* stands for ‘everything or nothing’. If any part of the transaction is left incomplete then the entire transaction is considered failed.
- *Consistency* ensures that a database before and after any transaction is stable at a valid state.
- *Isolation* ensures that multiple transactions executing at the same time do not affect one another's execution. Thus, requiring the concurrent transactions to be serialized.
- *Durability* ensures that once a transaction has been committed it will remain in the same state i.e. stored permanently even if there are some errors, or even if the system crash or power loss occurs.

When the set of data items are organized with the help of formally described tables the database formed is known as *relational database*. It can easily be accessed, created and extended. Because of this reason it is a predominant choice of storing data over hierarchical model or network model.

On the other hand, when traditional tables are not used to store data, it is known as non- relational database. These databases may store the data in the form of key- value stores, XML format, multidimensional databases and so on.

Non- relational databases, also commonly known as NOSQL(Not Only SQL), provide elastic scaling meaning that they scale up transparently by adding a new node, and they are usually designed with low-cost hardware in mind. The other advantages that come packaged with non relational databases are: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge amount of data and more.

This paper is organized as follows. Section 2 consists of a general overview of relational database system. Section 3 elaborates on a non-relational databases. Finally the results of the survey and comparison are discussed in section 4.

2. RELATIONAL DATABASE

2.1 Overview of Relational Database Model

E.F Codd invented the relational database in 1970. A database is an application that allows storing and retrieving data very rapidly. A relational database is a collection of data items organized in formally-described tables from which data can be accessed or reassembled in many different ways. Relational Database is a set of tables referred to as relation with data category described in columns similar to spreadsheets. Each row contains a unique instance of data for the corresponding data category. While creating a relational database domain of possible values along with constraints are applied to the data. It is the relation between the tables that makes it a 'relation' table. They require few assumptions about how data will be extracted from the database. As a result, the same database can be viewed in many different ways.

Mostly all the relational databases use Structured Query Language (SQL) to access and modify the data stored in the database. Originally it was based upon relational calculus and relational algebra and is subdivided into elements such as clauses, predicates, queries and statements.

Some of the benefits of the database designed according to the relational model are:

- Most of the information is stored in the database and not in the application, so the database is self documenting.
- It is easy to add, update or delete data.
- It gives benefits of data summarization, retrieval and reporting.
- The database is structured in a tabular form with highly related tables; the nature of the database is predictable.
- Also, any changes required to make in the schema of the database is quite simple.

2.2. Tools of Relational Database

The two most extensively used relational databases are MySQL and Oracle. MySQL is more popular with the websites. It is a light weight system which is extremely fast but Oracle is majorly used in case of large database requirement like Banking, Insurance, ERP and finance companies. It is used to solve complex problems and supports large OLTP environments.

Though they majorly work in a similar fashion yet there are a few differences between them.

- Oracle is a high cost system that can be afforded only by large organizations unlike MySQL which is open source database. But MySQL lacks on add on facilities and robustness supported by Oracle.

- Oracle has large tablespace, role management, snapshots, packages and synonyms unlike MySQL.
- The syntax of Oracle is high flexible. It provides integrated programming language like PL/SQL. It also has broader command structure as compared to MySQL.
- The security of Oracle is very tight and bounded. MySQL provides three security parameters like username, password and location whereas Oracle provides advance security enhancements by creating profiles, local authentication and external authentication.
- Oracle is case insensitive but MySQL is case sensitive when it comes to database and table names.
- For data transfer Oracle uses XML but this language is not supported by MySQL.
- MySQL has CHAR and VARCHAR for character types on the contrary Oracle has four character types namely CHAR, NCHAR, VARCHAR2, and NVARCHAR2 all types need to be minimum of 1 byte long.
- Oracle is a highly extensive database system that supports advanced functionality like Active Data Guard, Audit Vault, Partitioning and Data Mining. MySQL does not support Audit Vault on the server.
- Temporary tables are handled differently by Oracle and MySQL. In MySQL the tables are dropped automatically after the current session ends and are visible only to the current user session but in case of Oracle the tables have to be explicitly dropped. The tables are visible to all the sessions but the data inside is visible only to the current user session.
- MySQL has backup utilities like mysqldump and mysqlhotcopy. Oracle has Recovery Manager (RMAN) which is the most important backup utility. Using this backup scheduling can be automated using a very few commands and scripts.

Oracle has tons of features starting from user defined data types to database management tools extending upto XML. Because of its high functionality it has reduced the need of any third party software or tool. It supports large business application.

MySQL is used in case of high speed web and gaming. It supports small sized data warehouse and OLTP systems so it can be used in small projects in smaller companies at low cost. However it requires third party softwares and lacks the robust functionality provided by Oracle.

2.2. Shortcomings of Relational Database

Relational Databases though conventionally accepted suffer from a number of drawbacks which are discussed in this section

- 1.) Relational databases do not support high scalability, until a certain point better hardware can be employed but beyond that point the database must be distributed.
- 2.) One of the major disadvantage is data is stored in relational database in form of tables, this structure can give rise to high complexity in case data cannot be easily encapsulated in a table.
- 3.) Much of the features provided by relational databases are not used hence simply add to the cost as well as the complexity of the database.
- 4.) Relational Databases make use of SQL, which is featured to work on structured data, but SQL can be highly complex when working with unstructured data.
- 5.) When the amount of data turns huge the database has to be partitioned across multiple servers, this partitioning poses several problems because joining tables in distributed servers is not an easy task.

Thus, there was a demanding need to store and access huge data in a better way, thus giving rise to a newer form of data storage known as NON-RELATIONAL DATABASES.

3. NON-RELATIONAL DATABASES

3.1 Overview

Non relational database is a class of systems which manage databases and it broadly differs from the relational systems in many significant ways; most important being that it doesn't use relations (tables) as its storage structure. Other factors which differentiate it are it doesn't use SQL as its query language, join operations cannot be performed, it doesn't guarantee ACID properties and can be scaled horizontally. There can be many classification for NOSQL databases that are available today. One of the classification can be based on CAP theorem as is discussed in [1]. **Non relational databases may primarily be classified on the basis of way of organizing data as follows:**

1. Key Value Stores

It allows the app-developer to store schema-less data. This data consists of a key which is represented by a string and the actual data which is the value in key-value pair. The data can be any primitive of programming language, which may be a string, an integer or an array or it can be an object. Thus it loosens the requirement of formatted data for storage, eliminating the need for fixed data model.

2. Document Store

Document Store, also commonly known as "Document Oriented Database", is basically a computer program used for storing, retrieving, updating data stored in database.

The underlying storage structure used in such databases is a 'document'. Each Document Store differs in its implementation of data; however each of it assumes that data is enclosed and encoded in some standard format which may be XML, BSON, PDF or Microsoft office. Each document is represented by a unique key which is a string (URI or path). An API or a query language is provided for fast retrieval of documents on the basis of its content. For e.g. a query that retrieves all the documents in which certain field is set to some particular value.

3. Graph Database

Graph databases are schema-less databases which use graph data structures along with nodes, edges and certain properties to represent data. Nodes may represent entities like people, business or any other item similar to what objects represent in any programming language. Properties designate any pertinent information related to nodes. On the other hand; edges relate a node to other node or a node to some property. One can obtain some meaningful pattern or behavior after studying the interconnection between all three viz. nodes, properties and edges.

4. Column Oriented Databases

Column Store Databases, unlike Row Databases, store their data in the form of columns. It serializes all the values of one column together and so on. Column-oriented databases are comparatively efficient than row oriented one's when new values for a column are entered for all rows at once as column data can be written efficiently and replace old data without altering any other columns for the rows.

5. Object Oriented Databases

Object Oriented Databases also commonly known as OODBMS), is a database system. It stores its data in the form of objects. This feature supports inheritance and hence reusability similar as in object oriented programming.

6. Grid and Cloud Databases

Grid and Cloud Databases make use of grid and cloud computing collectively. Grid computing is exploited to manage heterogeneous and geographically distributed databases while Cloud Computing provides easy access to remote hardware and storage resources.

7. XML Databases

XML database is a database management system that is used to store XML data. In all native XML databases, XML document is the fundamental storage format.

However in some XML enabled databases, data of the XML document is divided into parts and these parts are stored within tables using another XML mapping layer. Some hybrid XML databases also exist which are combination of native and XED's.

8. Multidimensional Databases

Multidimensional Databases store the data as n-dimensional matrix. All the useful aggregates are also precompiled and stored, allowing roll-ups and drill-downs to be answered interactively. Many products use this approach, like, Arbor Essbase [Arb] and IRI Express [IRI]. Multidimensional databases may use a relational database as backend in which multidimensional queries are mapped onto equivalent relational queries. Products which use this concept are Redbrick and Microstrategy

9. Multivalue Databases

Multivalue database earlier known as PICK database is such a database which understands three dimension data directly i.e. **fields**, **values** and **subvalues**. In this, value is a breakdown of field and subvalue is further breakdown of value. The tables in this database are extremely flexible such that if any of the changes are made in the database, there is no need to shut down the database or rebuild the database. Also multivalue databases have calculated columns as they can contain small programs for calculations.

10. Multimodel Databases

Multimodel databases are a blend of various other non-relational databases that provides a mix of advantages offered by various other types mentioned above.

Non-relational databases provide a solution to the increasing amount of data storage required in the internet today. With the advent of NOSQL, cloud computing can now effectively manage the huge amount of data and its efficient handling. A new architecture known as CDSA(Cloud data storage architecture) [2] further improves query performance. In [5] the author shows performance improvement by using NOSQL instead of MS SQL in execution of a textbook management system.

Non Relational database can be classified in a way as shown in Table 1.1. As shown in the table document store, key value and corresponding XML databases are listed. A more elaborate description can be found in [6]. A comparative analysis of various NOSQL databases has already been done previously in [3] and [10].

<u>Core NoSQL Systems</u>	<u>Document Store</u>	<u>Key Value / Tuple Store</u>	<u>XML Databases</u>
Hadoop/HBase	MongoDB	DynamoDB	Mark Logic Server

Cassandra	CouchDB	Azure Table Storage	EMC document xDB
Hypertable	RavenDB	MEMBASE	EXist
Amazon Simple DB	Citrusleaf	Riak	Sedna
Cloudata	Clusterpoint Server	Redis	BaseX
Cloudera	ThruDB	LevelI DB	Qizx
SciDB	Terrastore	Chordless	Berkely DB XML
Stratosphere	SisoDB	GenieDB	
	SDB	Scalaris	
		Tokyo Cabinet/Tyrant	
		GTM	
		Scalien	
		Berkeley DB	
		Voldemort	
		Dynomite	

Table 1.1

3.2. Disadvantages of NOSQL

Apart from various advantages such as high throughput, horizontal scalability, avoidance of expensive object relational mapping, Non-Relational Databases have some shortcomings discussed in [4] and [7]. The most significant ones are discussed here.

- 1.) Most of the Non-Relational databases are open source software and though well appreciated, it compromises in reliability as nobody is responsible in times of failures.
- 2.) Many of the Non-Relational databases are disk-based which implement buffer pool and multithreading hence require buffer management and locking features which add on to performance overhead.
- 3.) Many Non-relational databases provide BASE properties and sacrifice conventional ACID properties as a step to increase performance. This could mean that non-relational databases compromise on consistency within the database.
- 4.) Because of flouting ACID properties, the degree of reliability provided by non-relational databases is lower than what is provided by the relational databases. Developers have to rope in programming to apply ACID restraints which could have been provided easily in relational databases.
- 5.) As the market is still unfamiliar to non-relational databases, it causes a major reason for developers

to use it and in determining whether it will be beneficial or not.

NoSQL solutions have emerged and risen because of deficiencies in existing RDBMS technology.

- **Data which is updated or written more often rather than being read** should consider using a **key-value store** like Redis, or a **document store** like MongoDB.
- **Data which is big** must try considering **distributed database** system like Hadoop.
- **Files which store data in binary form** (such as audio files etc) work greatly with a database that aids to user's browser, like Amazon S3.
- **Transient data** must be worked out with a transient datastore like Memcache.0
- **Data which needs to be replicated** to number of locations (eg. A database between a web application and a device like mobile) should use CouchDB.
- Applications which require high availability i.e. where **minimizing downtime is crucial**, must utilize the features of datastores like Cassandra and Riak which provide automatic clustering and redundant setup.
- **Data which needs to work with Graphs** must use Neo4J - built for traversing social graphs

4. Results and Discussions

Finally , we can conclude and highlight the major differences between the two types of databases as follows :

S.NO	NON- RELATIONAL	RELATIONAL
1.	High data throughput	Low data throughput
2.	Highly scalable	Less Scalable
3.	The data can be inserted anytime without defining a schema. The data can be altered anytime with any major issue. So its highly flexible	In this case the data has to fit in predefined tables or structure.
4.	Performance can be enhanced by caching data into system memory.	Caching has to be done with the help of special infrastructure.
5.	In the application multiple operation transactions can be implemented.	Transactional
6.	Single index, key value store	Index available on multiple column.
7.	Provide BASE properties	Provides ACID properties
8.	Non relational databases compromise on Consistency	Provide better consistency than Non-relational databases
9.	Data duplication is allowed which threatens data integrity. Moreover updating one record means updating all other duplicate records which means high overhead.	The relational database design process eliminates record duplication, preventing inconsistent data from occupying the database
10.	Searching in Non-relational database is inefficient especially those based on multiple criteria, as they require more than one pass.	Relational database organization allows a query language such as SQL to use the primary key shared between tables to quickly and efficiently collate and return the requested records

Table 1.2

References

- [1] Jing Han; Haihong, E.; Guan Le; Jian Du; , "Survey on NoSQL database", Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on , vol., no., pp.363-366, 26-28 Oct. 2011 doi: 10.1109/ICPCA.2011.6106531
- [2] Jing Han,Meina Song, Junde Song " A Novel Solution of Distributed Memory NoSQL Database for cloud Computing" , ICIS '11 Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science . Pages 351-355 ISBN : 978-0-7695-4401-4
- [3] Tudorica, B.G.; Bucur, C.; , "A comparison between several NoSQL databases with comments and notes," *Roedunet International Conference (RoEduNet)*, 2011 10th , vol., no., pp.1-5, 23-25 June 2011 doi: 10.1109/RoEduNet.2011.5993686
- [4] Okman, L.; Gal-Oz, N.; Gonen, Y.; Gudes, E.; Abramov, J.; , "Security Issues in NoSQL Databases," *Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011 IEEE 10th International Conference on , vol., no., pp.541-547, 16-18 Nov. 2011 doi: 10.1109/TrustCom.2011.70
- [5] Wei-ping Z. and Ming-xin L. "Using MongoDB to Implement Textbook Management System instead of MySQL", 3rd IEEE International Conference on Communication Software and Networks (ICCSN), May 27-29, 2011, Xi'an, China.
- [6] Stefan Edlich, "List of NoSQL Databases," July 2011. [Online]. Available: <http://nosql-database.org>
- [7] Neal Leavitt " Will NoSQL Databases Live Up to Their Promise?" IEEE Computer Society 0018-9162/10/\$26.00 © 2010 IEEE
- [8] C. Strauch, "NoSQL Databases," February 2011. [Online]. Available: <http://www.christof-strauch.de/nosql dbs.pdf>
- [9] Uma Bhat, Shraddha Jadhav, " Moving Towards Non-Relational Databases "©2010 International Journal of Computer Applications(0975 – 8887)Volume 1– No. 13
- [10] Clarence J M Tauro, Aravindh S, Shreeharsha A.B "Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases" International Journal of Computer Applications (0975 – 888) Volume 48– No.20, June 2012