e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

NOSQL DATABASE

Sudam Devram Dhasade[*1]

[*1]B.K. Birla Collage Kalyan, India.

ABSTRACT

NoSQL is a concept used to refer to non-relational databases. Thus, it encompasses the majority of the data stores that aren't based totally on the conventional RDBMS principles and are used for managing big data sets on an Internet scale. Some of the applications of web service 2.0 want large data handling. This needs the existing relational database to scale horizontally with a purpose to achieve demand for excessive performance, mainly for applications which require excessive scale of user data and of excessive concurrency. These problems are essential consideration for designers to come up with a new group of databases, popularly called as NoSQL. The developing demand for cloud computing and the improvement of Internet motivates the NoSQL movement. This paper deals with features and data models of NoSQL databases utilized in cloud computing environment along with power and limitation of every of the model. In addition, this paper talks approximately classification of NoSQL databases based upon CAP theorem.

Keywords: NoSQL; RDBMS; CAP Theorem; Document Oriente; Column Family; Big Data, JSON.

I. INTRODUCTION

Carl strozz introduced the concept of NoSQL in 1998. NoSQL stands for Not Only SQL. Storing distributed data is the major purpose of using a NoSQL database. NoSQL database is simply an alternative to traditional relational database. The industry of database has seen an introduction of many non relational databases such as MongoDB, Hbase, Neo4j in previous couple of years. Depending upon the commercial business requirement and approach a cloud dealer can go together with any of the database type. Still some designers of pre relational database claim the NoSQL databases now no longer to be Capability sufficient in managing data integrity. This paper is prepared as follows: Section two describes the significance of NoSQL databases. Section three highlights at the NoSQL data models. Section four highlights at the transaction in NoSQL databases. Section five places mild at the comparison for NoSQL databases. Finally, we finish this paper in Section 6 with conclusion.

II. IMPORTANTCE OF NOSQL

A. Background-

For the last few years, SQL vs. NoSQL (Not SQL) has been emerged as a heated argument over the Internet. The argument "SQL vs NoSQL," really talks concerning relational versus non-relational databases. Due to normalized data model and enforcement of strict ACID properties, traditional relational database is taken into account to be a schema based transaction oriented database. It requires a strict predefined schema prior to storing records into it. Redefining a schema in case of a future change, once after data were given inserted into the database is disruptive. Whereas within side the generation of Big Data, there's a constant need for including new kinds of data to enhance the applications. Again, the storage answer of relational database can make a large impact on velocity and scalability. Web offerings like Amazon and Google have terabytes and petabytes of data saved of their big data centers and have to reply to huge read-write requests without a major latency. To scale a relational database, data needs to get disbursed on multiple servers. Before supplying to the software, the desired data has to be collected from many tables and combined. Similarly, at the same time as writing facts also; it has to be completed on many tables in a coordinated manner. For any software, it may be a bottleneck to manage tables across multiple servers. In relational databases, "join" operation slowdowns the system to a crawl, especially when hundreds of thousands of users are doing lookups in opposition to tables with hundreds of thousands of rows of data. Large scale web services including Google, Amazon, Yahoo, Facebook located those to be the cases to develop their personal non-relational database with a view to meet the scalability and performance needs.

B. Features of NoSQL-

NoSQL databases won't require a predefined table schema, generally scale horizontally and commonly avoid join operations. Due to schema less nature and involvement of smaller subset analysis of NoSQL system, this

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

database can be higher defined as structured data stores. Three essential simple functions of NoSQL databases are scale-out, flexible data structure and replication, that are defined as follows.

- Scale-out: Scaling out refers to attain excessive performance in a dispensed environment by the use of many general-motive machines. NoSQL databases permit the distribution of the data over a massive number of machines with a dispensed processing load. Many NoSQL databases permit automatic distribution of data to new machines while they may be added to the cluster. Scale-out is evaluated in phrases of scalability and elasticity.
- Flexibility: In phrases of data structure says that there's no want to define a schema for databases. NoSQL databases don't longer require a predefined schema. This permits the users to store data of different structures in the equal database table. However, help for high-level query languages which include SQL isn't supported through maximum of the NoSQL databases.
- Data Replication: One of the most important features of NoSQL databases is data replication. In this method, a duplicate of the data is distributed to distinct systems in order to attain redundancy and load distribution. However, there may be a chance of losing information consistency in many of the replicas. But it is believed that every so often this consistency can be achieved eventually. Consistence and availability are the elements for evaluating replication.

III. NOSQL DATA MODELS

There are four main categories of NoSQL database models such as, Key-Value Data Stores, Document Oriented Data Stores, Column Family Data Stores and Graph Database.

A. Key-Value Data Stores :-

In order to handle incredibly concurrent access to database, the category of NoSQL designed is key-value stores. It is the simplest, still the maximum effective data store. In a key-value store, every data includes a pair of a completely unique key and value. In order to keep data, a key receives generated through the software and value gets related to the key. And this key-value pair receives submitted to the data store. The data values saved in key-value stores may have dynamic sets of attributes attached to it and is opaque to the database control systems. Hence, key is the simplest way to access the data values. The kind of binding from the key to value relies upon at the programming language used with inside the application. An application desires to offer a key to the data stores with the intention to retrieve data. Many key-cost facts shops use a hash function. The application hashes the key and discover the area of the data with inside the database. The key-value data stores are focused on row. Which means it allows the application to retrieve data for entire entities.

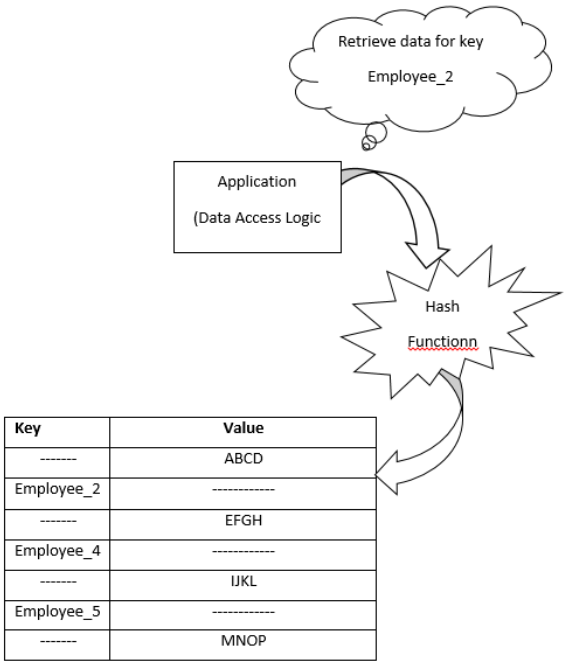

Fig 1. Example of a key-value data store

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

As shown in Fig.1, describes retrieval of data from a key-value database. The application has certain a key 'Employee_2' to the data store, a good way to retrieve data. Using the hash feature, the application hashes the key, a good way to trace the location of data with inside the data store. The layout of the key should support the maximum common queries fired at the data store. Efficiency of the hash function, layout of the key and size of the values being saved are the elements which affect the overall performance of a key-value data store.

B. Document Oriented Data Stores :-

At an abstract level, a document oriented database is much like a key-value data store. It additionally holds value, which an application can study or fetch through the usage of a key. Several document databases robotically generate the particular key while creating a new document. A document in a document database is an entity, that's a set of named fields. The feature which distinguishes the document oriented database from a key–value data store is transparency of the data held through the database. Hence, the query possibility isn't limited with the key only. In order to support scenarios where the application needs for querying the database not only based on its key however additionally with attribute values, can change for document databases. A document wishes to be self-describing in a record oriented database. Information is saved in a portable and properly understood layout which include XML, BSON or JSON.

As shown in Fig. 2, the document database stores records in the form of key-value pairs. But the records stored with inside the database is transparent to the system, not like key-value databases. The application can query the database now no longer only with the key i.e. 'Employee ID' but additionally with the defined fields with inside the record i.e. FirstName, LastName, age etc. Document records stores are efficient approach to model data based on common software program problems. But it comes on the price of barley decrease performance and scalability in contrast to key-value records stores. Few of the maximum prominent record stores are Riak, MongoDB, CouchDB.

Key (Employee ID)	Document
Employee_1	FirstName:Alex LastName:Roy Age:28 .....
----------	----------
Employee_3	FirstName:Jeson LastName:Stark Age:35 .....
----------	----------
Employee_4	FirstName:Jonson LastName:Mathew Age:23 .....
----------	----------

Fig 2. Example of Document Oriented Data Stores.

C. Column Family Data Stores :-

Sometimes an application may need to read or fetch a subset of fields, much like the SQL's projection operation. Column family data store permits storing data in column centric approach. The column family data store partitions the key area. In NoSQL, a key area is taken into consideration to be an item which holds all column families of a layout together. It is the outer maximum grouping of the data with inside the data store. Each partition of the key area is understood to be a Table. Column families are declared through those tables. Each column family includes a number of columns. A row in a column family is based as collections of arbitrary number of columns. Each column is a map of a key-value pair. In map, keys are the names of columns and columns themselves are the values. Each of those mappings is known as a cell. Each row in a column-family database is recognized through a unique row key, defined through the application. Use of those row keys makes the data retrieval quicker. In order to avoid overwriting of the cell values, of few of the famous column-family databases add timestamp data automatically to individual columns. Every time there may be an update, it

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

creates a new edition of the cells that have been affected by the update operation. Always, the reader reads the value that is last written or committed. A row key, column family, column and timestamp represent a key. Hence the precise mapping may be represented as

(row key, column family, column, timestamp)- > value.

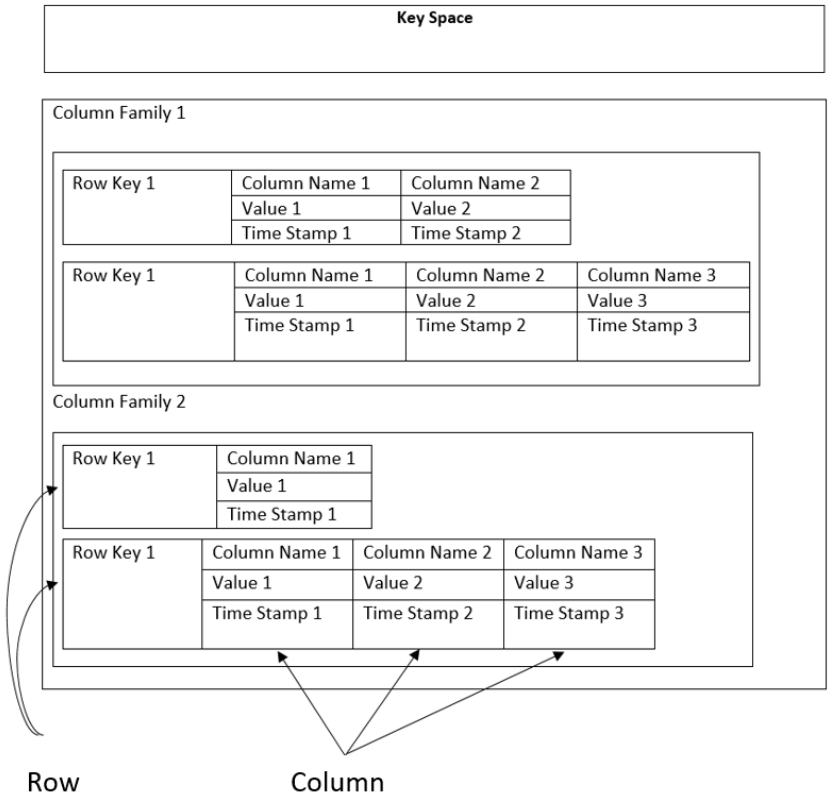

Fig 3. Example Column Family Data Store.

This data model has been popularly common as "sparse, disbursed, consistent multidimensional sorted map". A benefit of the usage of a column family data store over a conventional database is in managing NULL values. In a relational database, while a value for an attribute isn't always applicable for a specific row, NULL receives stored. While in a column family database, the column may be simply removed for the corresponding row in case the data isn't always available. That why Google calls it a sparse database. One of the key features of this database is that it could be disbursed in a billion of cells over thousands of machines. The cells are sorted on foundation of row keys. Sorting of keys permits searching data for a range of keys. Since the data in such type of model get organized as a fixed rows and columns, representation wise this database is maximum much like the relational database. But like a relational database, it does not want any predefined schema. At runtime, rows and columns may be added flexibly, but oftentimes the column families must be predefined, which leads the data store to be much less flexible than key-value or document data stores. Developers must understand the data captured by the application and the query possibilities earlier than deciding the column families. A well-designed column-family database enables an application to satisfy the majority of its queries by visiting much less number of column families as possible. Compared to a relational database keeping equal amount of data, a column family data store is extra scalable and faster. But the overall performance comes on the rate of the database being much less generalized than a relational database, as it's far designed to assist for a particular set of queries. HBase and Hypertable database systems are primarily based totally on the data model described above. Whereas any other database system, Cassandra differs from the data model, as it is having a new dimension added known as super column. As shown in Fig. 4 a super column includes more than one column. A collection of super columns along with a row key constitute a row of a super column family. As in columns, the super column names and the sub column names are sorted. Super column is likewise a name-value entity, but without a timestamp.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

D. Graph Database :-

Graph databases are considered to be the experts of highly linked data. Therefore, it handles data concerning many relationships. There are typically 3 core abstractions of graph database. These are nodes, edges which join different nodes, and properties. Each node holds data about an entity. The edges represent the life of relationship among the entities. Each relationship is having a relationship kind and is directional with a beginning point (node) and an end point. The end poin may be a few other nodes than that of the start node, or possibly the same node. Key-value properties are related not only with the nodes, but also with the relationships. The properties of the relationships provide extra information about the relationships. The direction of the relationship determines the traversal path from one node to the alternative in a graph database.

Fig. 4 represents part of the 'Employee' database established as graph database. Each node on this graph database represents an employee entity. These entities are associated with every different through a relationship of relationship type “knows”. The assets related to the relationship is “Duration”. The key distinction between a graph and relational database is data querying. Instead of the usage of fee extensive process like recursive join as in relational database, graph databases use traversal method. While querying through graph database, a beginning node needs to be specified by the application. Traversal starts from the start node and progresses through relationships to nodes related to the beginning node, primarily based totally upon a few rules described by the application logic. The traversal technique involves only nodes that are relevant to the application, not the entire data set. Hence, a massive growth in quantity of nodes does not have an effect on the traversal rate much. Social networking, data mining, handling networks, and calculating routes are a few of the fields in which graph database has been used extensively. Neo4j, GraphDB are famous graph databases in use today.

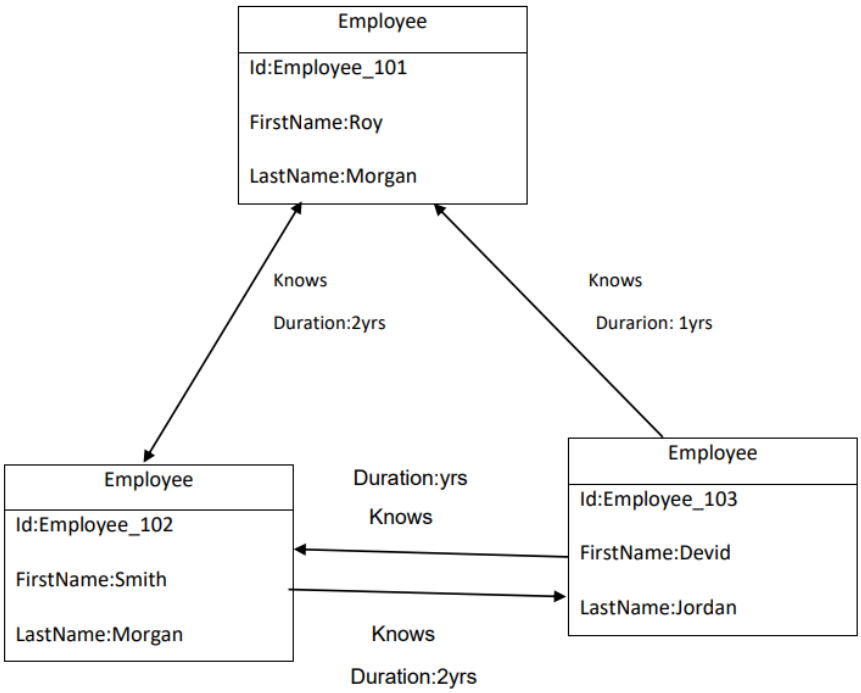

Fig 4. Example of graph database

IV. TRANSACTION IN NOSQL DATABASES

When we speak about SQL vs. NoSQL, the competition is really not between the databases. The comparison is between the transaction models of each of the databases. Transaction is defined to be the logical unit of a database processing formed through an executing program. The transaction of SQL database is primarily based upon strict ACID properties. Where ACID is the abbreviation for Atomicity, Consistence, Isolation and Durability. But designers of the NoSQL database came up with a decision, that ACID property is simply too restrictive to obtain the needs of big data. Hence, Professor Eric Brewer with inside the year of 2000 came up with a new theorem called as CAP theorem. CAP is the abbreviation for Consistency, Availability and Partition

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

tolerance. The theorem says that the designers can obtain any of those properties at a time in a disbursed environment. The designers can make sure Consistency and Availability at the cost of Partition tolerance, i.e. CA based database. If the designer is going for availability and partition tolerance at the cost of Consistency, then it's far an AP based database. And if to make sure Consistency and Partition tolerance at the cost of availability, than the database is CP based. The transaction of NoSQL may be classified as follows.

- Concerned about consistency and availability (CA):This type of database system ensures its priority more towards data availability and consistency by the use of replication approach. Part of database doesn't trouble about partition tolerance. In case of incidence of a partition between nodes, the data will exit of sync. The relational database, Vertica, and Greenplum database systems fall under such class of databases.
- Concerned about consistency and partition tolerance (CP): The precedence of such database system is to make sure data consistency. But it does now no longer support for good availability. Data receives stored in distributed nodes. When a node is going down, data becomes unavailable to hold consistency between the nodes. It keeps partition tolerance by preventing resynchronization of data. Hypertable, BigTable, HBase are few database systems that are concerned about CP.
- Concerned about availability and partition tolerance (AP): The precedence of such database system is to make sure data availability and partition tolerance primarily. Even if there's a conversation failure between the nodes, nodes stay online. Once after the partition receives resolved, resynchronization of data takes place, but without the assured of consistency. Riak, CouchDB, KAI are some databases which follow this principle.

Afterwards, CAP theorem gets extended into PACELC. PACELC is stand for partition, availability, consistency, else, latency, consistency. According to this model, the tradeoff between availability and consistency isn't only primarily based totally upon partition tolerance, but it's also dependent on the existence of network partition. It suggests latency to be one of the crucial factors, since maximum of the disbursed database systems use replication technology for making sure availability. Later, eBay introduced a new theorem called as BASE theorem. BASE goals to achieve availability instead of consistency of databases. BASE is the abbreviation for typically available, soft state and finally consistent.

- Basically Available: Basically to be had says that despite the fact that part of the database turns into unavailable, different elements of the database hold to characteristic as expected. In case of a node failure, the operation keeps at the reproduction of the information saved in a few different nodes.
- Soft State: Soft state says that on the basis of user interaction, a data can be depending on time. These data may also have possible expiration after a certain duration of time. Hence, to hold the data relevant in a system it has to be updated or accessed.
- Eventually Consistent: Eventual consistency says after any data update, data won't become consistent throughout the entire system but it will become consistent with time eventually. Therefore, the data is stated to be consistent within side the future.

V. COMPARISION OF NOSQL DATABASES

There isn't any difficult and fast rule to decide which NoSQL database is great for an enterprise. Business Model, strategy, cost and transaction model demand are few of the crucial elements that an enterprise need to consider while selecting a database. Following are a few of the facts which may assist in selecting a database for an enterprise.

- If the applications simply store and retrieve data objects that are opaque to the database management system and blobs by the usage of a key as identifier, then a key-value store is the great choice. But if the application likes to query the database with a few attribute values other than the key, it fails. Also, while updating or reading an individual field in a document key-value store is a failure.
- When applications are extra selective and want to filter records primarily based totally on non-key fields, or retrieve or update individual fields in a record as it, then document database is a good solution. Document data stores provide higher query possibility than key-value data stores.
- When the applications want to store data with hundreds or thousands of fields, but retrieves a subset of these fields in maximum of the queries that it performs, if so column-family data store is an efficient choice. Such data stores are appropriate for big datasets that scale high.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

- If the applications want to store and process data on closely linked data with highly complex relationship between the entities, graph database is the great choice. In a graph database, entities and relationship between the entities are treated with equal importance.

Table 1 represents a listing of databases, their corresponding data models, alongside transaction model and query language utilized by those databases. Cassandra for facebook, HBase for Google, DynamoDB for Amazon are few of the databases which had been developed by distinct companies in order to meet their demand for excessive data storage requirement. On the other hand, database systems including Neo4j, Riak, and MongoDB had been developed in order to serve other organizations. In phrases of transaction model, maximum of the databases including DynamoDB, Riak, Cassandra and Voldermort give extra preference to availability over consistency. Whereas Tokyo Cabinet, Hbase select consistency over availability. NoSQL database was designed in order to manage huge volume data processing, excluding a number of the support system of RDBMS like ad-hoc query. Though, a lot of NoSQL databases mentioned in Table. 1 support ad-hoc queries, but the stage of programming knowledge in writing queries needs to be much better than that of a relational database.

Table.1. Comparision of Different NoSQL Databases

Database Tool	Data Model	Transaction Model (CAP)	Ad-HOC query
DynamoDB	Key-value	Availability and Partition Tolerance	Built in API
Riak	Key-value	Availability and Partition Tolerance	CorrugatedIro N
Voldermort	Key-value	Availability and Partition Tolerance	No
Tokyo Cabinet	Key-value	Partition Tolerance and Consistency	No
CauchDB	Document	Availability and Partition Tolerance	Cloudant, Lucene
MongoDB	Document	Availability and Partition Tolerance	BSON based format
RavenDB	Document	ACID	Built in, Limited
Cassandra	Column-Family	Availability and Partition Tolerance	HIVE, PIG
Hbase	Column-Family	Partition Tolerance and Consistency	HIVE, PIG
Neo4j	Graph	Consistency and Availability	Chyper

VI. CONCLUSION

In the database domain, the NoSQL database is taken into consideration to be quite new. However, those are being developed on recognized and existing theory. NoSQL databases systems still have diverse limitations. There is not a common standard or not any common and familiar query language for querying NoSQL databases. Each database behaves uniquely and does things differently. Relatively, those databases are immature and constantly evolving. NoSQL database does not assist strict ACID properties, hence there's no assure that all data could be written successfully to the data store. This paper describes the limitation of relational database alongside unique categories of NoSQL data models. Since there may be no evaluation available to discover the proper tool, this paper compares the power and limitation of every the data model. Limitations of NoSQL databases and its use in a cloud computing environment are the regions which want detailed research in the future.

VII. REFERENCES

[1] Jing Han, Haihong E, Guan Le,Jian Du, "Survey on NoSQL Database," IEEE, pp. 363- 366, 2011

[2] R Hecht, S Jablonski, "NoSQL Evaluation," International Conference on Cloud and Service Computing, IEEE, pp. 336-338, 2011.

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

[4] NoSQL Explanation by Datastax Academy:http://www.planetcassandra.org/what-is-nosql/

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

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

[7] Mahdi Negahi Shirazi,Ho Chin Kuan,Hossein Dolatabadi, "Design Patterns to Enable Data Portability between Clouds‟ Databases," 12th International Conference on Computational Science and Its Applications, pp. 117-118, 2012.