

**Athlone Institute of Technology**

**Report**

**The Implications of selecting a NoSQL rather than a relational database**

**Database 4**

*Chhaya Sharma*

*A00268860*

*Bachelor of Engineering (Honours) Software Engineering*

Table of Contents

[**1. Relational Database** 3](#_Toc36845761)

[**1.1Overview:** 3](#_Toc36845762)

[**1.2Advantage of Relational Database:** 4](#_Toc36845763)

[**2.No SQL Evolution:** 5](#_Toc36845764)

[**2.1 Support for Unstructured Data**: 5](#_Toc36845765)

[**2.2 High Scalability** 5](#_Toc36845766)

[**2.3 Low Operating Costs:** 5](#_Toc36845767)

[**2.4 Reduced Management:** 5](#_Toc36845768)

[3.**No SQL vs Relational Database**: 6](#_Toc36845769)

[**3.1 Implications of using NoSQL over Relational** 6](#_Toc36845770)

[**3.2 No SQL Approaches**: 7](#_Toc36845771)

[**4.** **RDBMS or NoSQL – Which one should Business go for?** 8](#_Toc36845772)

[**Reference:** 10](#_Toc36845773)

[**STUDENT PLAGIARISM DISCLAIMER FORM** 11](#_Toc36845774)

[PLAGIARISM DISCLAIMER 11](#_Toc36845775)

[Signed: Chhaya Sharma 11](#_Toc36845776)

## 1. Relational Database

### **1.1Overview:**

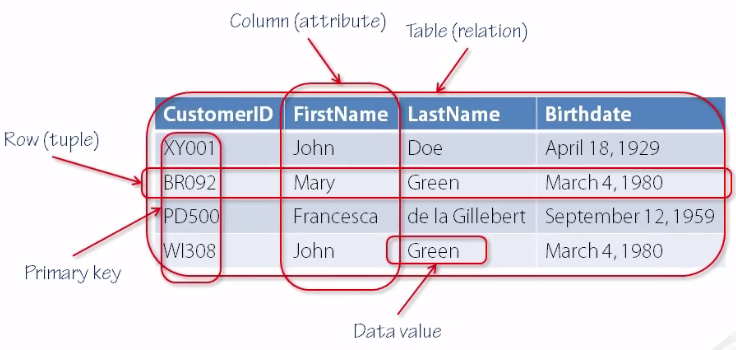
Data is the important key that drives the business and to manage and store this data we need to store it efficiently. Database has easily replaced traditional file system as with database we can easily store, organizes and manage large amount of data which was a challenging task in file system. As every organization whether big or small has data, it is important to choose appropriate database for the system.

In 1969, E.F invented a relational model which stores data in form of rows and columns. Relational Database is also based on same approach i.e. data is stored in the form of***tables or relations****.*

* Each table contains one or more data category in the form of **columns** or **attributes**.
* Each **row** contains some record or data hence is also called **record** or **tuple**.

Each table has a unique field (mostly id) called **Primary key**, this key plays a significant role in extracting records from the table.

In relational database, relationship between tables can be easily managed with the help of **Foreign Key.** Foreign key is column that act as a link to primary key of another table.



### **1.2Advantage of Relational Database:**

The relational databases are widely popular database among the alternatives like Object-oriented, hierarchical model and Flat file etc. Here are following reasons that make Relational Database such a preferred choice:

1. **No Redundancy**: In RDBMS, *Normalization* can reduce redundancy.
2. **Data Consistency:** RDMBS is best at ensuring data consistency across applications and database copies (instance). For example: If a customer deposit some money from bank branch updated balance should immediately reflect into his banking app and the website.

This feature of RDBMS makes it still a popular choice when it comes to critical business operations.

1. **Simple Structure**: In relational database data is a stored in tabular form which can be easily understand by the users who has some experience of working with spreadsheets and registers or any tabular data. This simplifies the development and use of database.

In RDBM data is highly structured and easily readable.

1. **Structural Independence:** With RDBMS, internal structure of the data can be easily change while keeping the data and information same so to change the entire database for changing the structure.
2. **Ease of Retrieval:** In relational database, data is easily accessible as we don’t need to navigate through some tree structure or hierarchy. User can query any table to create, read, update and delete records in table. Also, related tables can be combine using JOIN.

User can filter the data on basis of column content, allowing users to retrieve meaning information.

1. **Data Integrity:** Maintaining and assuring that data is consistent and correct. It is a necessary feature of relational model. Relational model allows user to force some constraints on the data, no operation can violate them. This provides strong data typing & This helps in ensuring that data falls in acceptable range.
2. **Flexibility:** Relational model is highly amendable as there is no limitation on how many tables you can have in a database how many rows and columns in a table etc. All these is dependent only on hardware and relational management system.

These models are flexible enough to adapt to changing requirements.

1. **Query Capability:** Use of query in accessing data is one of key reason of popularity of RDBMS. Structured Key Language or SQL is closely associated with RDBMS. Also, these queries are easy to use and allows users to specify what data they want to insert, update or delete etc. and they can get all level of detailed information using SQL queries.

# **2.No SQL Evolution:**

Relational databases are still extensively used database. But these databases can handle only a finite amount of data and in structured format. But in today’s world, enterprise and organizations has tremendous amount of data which is in unstructured format and can not be stored in table format which makes relational database incapable of handling this data. Also, relational database requires high maintenance which can be challenging when it comes to organizing big data.

To store and organize large amount of unstructured data, No SQL database were introduced. No SQL database has following merits over Relational database:

### **2.1 Support for Unstructured Data**:

Majority of data produced in the enterprises is unstructured. This feature of No SQL to manage unstructured text and information helps organizations in managing information for example any company can get reviews about a product in form of comments from user which in the form of text. With the help of No SQL these feedbacks can be easily stored, and decisions can be made.

**2.2 High Scalability**:

No SQL database is highly scalable as nodes can be easily scaled outward. The horizontal scaling in No SQL database is much cheaper, efficient and flexible then vertical scaling.

### **2.3 Low Operating Costs:**

No SQL uses *Commodity servers* which enables us to store large amount of data at low cost which makes it preferred choice over RDBMS because RDBMS servers were expensive, and their maintenance was quite a challenge.

### **2.4 Reduced Management:**

Most of No SQL database have automatic repair functionality hence it is quite easy to install and maintain them.

Big volume of data was produced from enterprises, organizations, browsing, e-commerce websites and companies at a rapid rate also this data is in unstructured format all these things were big challenge for RDBMS but were easily addressed by No SQL and hence Big data became the prime factor behind the popularity of No SQL.

# 3.**No SQL vs Relational Database**:

## **3.1 Implications of using NoSQL over Relational**

**Transaction Reliability:**

Relational Model, we get four fundamental properties as ACID (Atomicity, Consistency, Isolation, Durability). This make a transaction more reliable.

While with No SQL transactions are not that reliable as No SQL properties are BASE (Basically available, Soft state, Eventual consistency)

**Data Model:**

RDBMS stores data as tables/ relations in the form of tuples and attributes (rows and columns) i.e. structured format. This format is well organized and specific.

In NoSQL data is stored using modelling techniques e.g. Key value stores, graph and document model.

**Scalability:**

In Relational Database scalability is vertical means more hardware resources while NoSQL depends upon horizontal scalability (i.e. adding more commodity nodes) which is its biggest advantage over RDBMS as horizontal scalability is cheaper as compare to Vertical Scalability.

**Cloud:**

Since cloud database are not ACID properties compliant and they focus to provide better availability, scalability and flexibility along with support for structured, semi-structured and unstructured data hence NoSQL is best solution for them.

On the other hand, Relational database have rigid structure which is not suitable for Cloud.

**Big Data handling:**

As RDMS can handle finite amount of data and it need to be structured which is impractical when it comes to data produced in big organizations so NoSQL is a preferred choice in this case as it provides improved scalability and handling.

# **3.2 No SQL Approaches**:

**Key-Value Stores:**

A Key- value store, or key-value database uses associative array (e.g. map or dictionary) where each key is associated to only one corresponding value in a collection which is called Key-value pair.

* These key-value stores don’t have query language. Data is accessed and managed using get, put and delete commands
* Simplicity of this model makes it fast, scalable, flexible and easy to use.
* Here are some popular Key-stores:
  + Aerospike
  + Apache Cassandra
  + Riak
  + Redis
  + Couchbase Server

**Document Oriented Data Stores:**

This type of database uses document-oriented model to store data. These document stores store data in a single document only. These documents contain semi-structured data which can be queried using various query and analytical tools of DBMS.

* These can be a good choice of database if an organization wants to store different attributes along with large amount of data.
* These databases can scale horizontally. The data can be stored over thousands of computers but still it will perform good.
* Document Oriented Data Stores can be useful in blogging platforms, e-commerce applications, tweets, comments, ratings etc.

**Graph Database:**

These database uses graph data model which comprises of vertices and edges where vertices is an entity (e.g. person, car, place etc.) and edges are relationship between two nodes.

* Graph databases are useful because they show the links between relevant data.
* Few examples of best Graph databases are GraphDB Lite, Neo4j Community Edition, MapGraph etc.

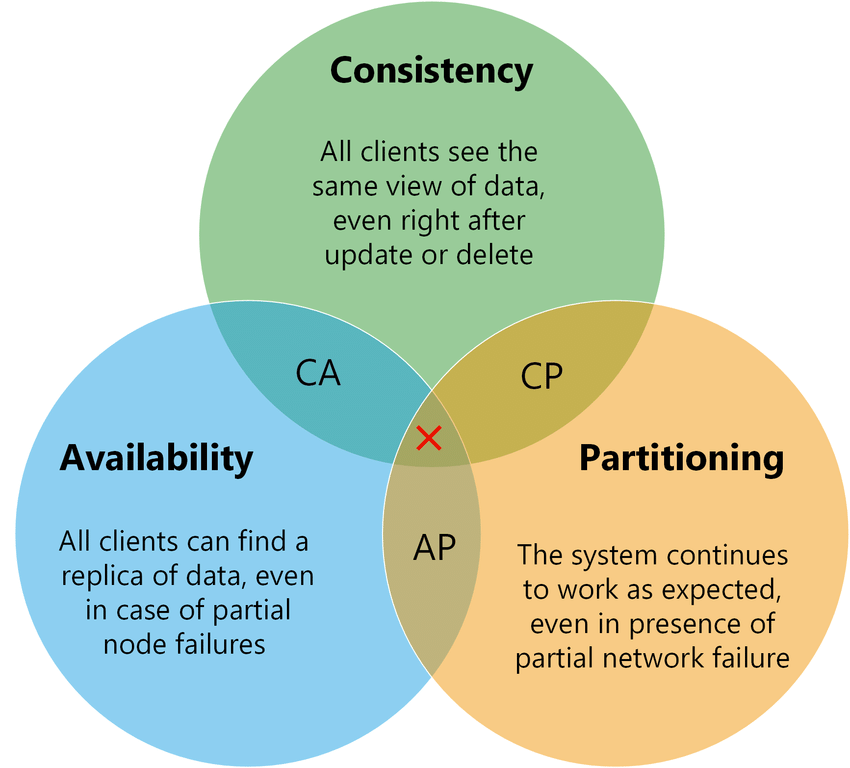
**Big Table Clones:**

Big Tables are peculiarly designed for large volumes of data, read and write performances and high availability. These databases have complex and richest data model and have more scalability in numbers of columns. Complex data is grouped into a single table.

* In big tables a table consists of groups of columns called families and a row key. These together enable fast lookup of a record of data in Bigtable.
* Using row keys: A row key is a unique string used as a reference a single record in big table. Used by Bigtable servers to equally distribute records between servers.
* Using Column Families is logical grouping of columns
* Using Timestamps: Different versions store in a column can be referenced using timestamps. Every time a different value is added with different timestamp.

# **RDBMS or NoSQL – Which one should Business go for?**

As now, both RDBMS and NoSQL are potential choices for any business. Choosing the right database is very important for any organization as it is directly linked with its success. Thus, the characteristics of databases must be heavily explored before deciding upon the correct database. These characteristics are given by CAP Theorem which explains Consistency, Availability and Partitioning.



For database as a cluster of nodes:

1. Consistency - Same data is available in every node thus forming a consistent system.
2. Availability - Cluster is available all the time for read and write. Even if one node fails, other can handle the request.
3. Partitioning - Even in case of node failures, system works perfectly. Other nodes are guaranteed to work as expected.

**RDBMS – ACID**

RDBMS like SQL focuses more on consistency and follows ACID properties.

1. **A**tomicity - Entire transaction takes place or nothing happens at all
2. **C**onsistency - Database must be in consistent state before and after the transaction.

**I**solation - Multiple transactions can occur independently without any interference with each other.

1. **D**urability - State of successful transactions are saved and can be recovered in case of system failure.

**NoSQL - BASE**

NoSQL gives more focus on availability and partitioning and provides eventual consistency. They follow BASE properties.

1. **Ba**sically Available - System does guarantee availability, in terms of the CAP theorem.
2. **S**oft state - State of the database may change over time to achieve eventual consistency.
3. **E**ventual consistency - System will become consistent over time.

Looking at the above comparison and properties, it is evidently clear now that both RDBMS and NoSQL have their own pros and cons. If the organization wants more consistency, they should go with RDBMS.

On the other hand, if availability is more of a priority than consistency or if the organization wants to support large no of users without compromising their demands, NoSQL is the one for them.

**Selection – IoT Company vs Bank**

Let’s consider the case where we need to select the best database for two organizations – IoT company and Bank based on the following data characteristics:

**Scalability**

RDBMS supports vertical scalability while NoSQL supports horizontal scalability

For new IoT company, going for database with scope for future expansion is best choice and hence they should go with NoSQL. Bank can go with either of them as they are rich.

**Data Retrieval**

Data retrieval is faster in NoSQL as they store data in unstructured format. RDBMS stores data in table that are often joined. This makes data retrieval in RDBMS time consuming.

IoT company has unstructured data coming from various sources and requires faster data retrieval. Thus, they should go with NoSQL.

Bank must have data in organized manner and accuracy is more of a concern than fast retrieval. Bank should go with RDBMS.

**System Maturity**

SQL supports key features like Security, Confidentiality, Authentication & Integrity. NoSQL doesn’t have such features and is more vulnerable to security breaches.

Bank must go with RDBMS as security is one of their main concern.

In IoT communication, security is required to transfer sensitive data, it is better for them to go with more secure RDBMS.

**Consistency & Availability**

Bank requires data to be in consistent state in the database. If the data is in consistent state, user may get different amount during different transaction. They should go with RDBMS.

IoT Company needs to handle varying no of users and speed requirements and hence partition tolerance is the priority here. They should do with NoSQL.

### **Reference:**

[1] Burtica, R.; Mocanu, E.M.; Andreica, M.I.; Tapus, N., "Practical application and evaluation of no-SQL databases in Cloud Computing," *Systems Conference (SysCon), 2012 IEEE International* , vol., no., pp.1,6, 19-22 March 2012

[2] Rautmare, S; Bhalerao D.M.; “MySQL and NoSQL database comparison for IoT application” *IEEE International Conference on Advances in Computer Applications (ICACA)* 24 Oct. 2016

**FORM A1**

# **STUDENT PLAGIARISM DISCLAIMER FORM**

## PLAGIARISM DISCLAIMER

STUDENT NAME: Chhaya Sharma

STUDENT NUMBER: A00268860

PROGRAMME: Bachelor of Engineering (honours) Software Engineering

YEAR: 2019-2020

MODULE: Database

LECTURER: Shelia Fallon

ASSIGNMENT TITLE: The Implications of selecting a NoSQL rather than a relational database

DUE DATE: 03-4-2020

DATE SUBMITTED: 03-4-2020

ADDITIONAL INFORMATION: Final report of NoSQL

I understand that plagiarism is a serious academic offence, and that AIT deals with it according to the AIT Policy on Plagiarism.

I have read and understand the AIT Policy on Plagiarism and I agree to the requirements set out therein in relation to plagiarism and referencing. I confirm that I have referenced and acknowledged properly all sources used in preparation of this assignment. I understand that if I plagiarise, or if I assist others in doing so, that I will be subject to investigation as outlined in the AIT Policy on Plagiarism.

I understand and agree that plagiarism detection software may be used on my assignment. I declare that, except where appropriately referenced, this assignment is entirely my own work based on my personal study/or research. I further declare that I have not engaged the services of another to either assist in, or complete this assignment.

## Signed: Chhaya Sharma

**Dated:** 03-4-2020