**Research Document**

S3-CB04

Web Shop – “**S. H. A. I. N.”**

Text

Description automatically generated

**Name :** Shanessa Kostaman (450080)

**Student Number :** 4082419

**Email :** [genoveva.shanessakostaman@student.fontys.nl](file:///C:\Users\shanessa\OneDrive%20-%20Office%20365%20Fontys\Semester%202%20-%20Software\PRJ\s-cb04-s2-g3-prj\genoveva.shanessakostaman@student.fontys.nl)

**Teachers :**

Jacco Snoeren – [j.snoeren@fontys.nl](mailto:j.snoeren@fontys.nl)

Onno Marsman – [o.marsman@fontys.nl](mailto:o.marsman@fontys.nl)

**Fontys University of Applied Sciences**

**Eindhoven – Netherlands**

**2020-2021**

Table of Contents

[1 What is my application about? What kind of data that I have? 3](#_Toc93087775)

[1.1 Comparing database statistics 3](#_Toc93087776)

[1.2 The different based on database type 4](#_Toc93087777)

[1.3 When to use relational database 6](#_Toc93087778)

[1.4 When to use non-relational database 6](#_Toc93087779)

[2 Conclusions 7](#_Toc93087780)

[3 References 8](#_Toc93087781)

# What is my application about? What kind of data that I have?

My application is about Online Web Shopping, on the Online Web Shopping we store all the products on the database to keep track on it. To check when the product’s finished, when the product has been sold. We also keep track our employees and members with database. Online Shopping is between user and product, which is mean the user has to be connected with the product.

## 1.1 Comparing database statistics

Based on the website that I visited, it says that MySQL is the most common database that has been used for the last 12 month in 2020.

Graphical user interface, text, application

Description automatically generated

From the graph MySQL Workbench is in the top 3 most use to communicate with the database.

Graphical user interface, application

Description automatically generated

## The different based on database type

There’s 2 type of database, which is non relational database and relational database. Here’s the different between both of them.

Relational Database:

* The relational database is properly follow a concept of transaction that refers to Atomicity, Consistency, Isolation, and Durability in a database system. The function of this properties is to make sure the accuracy and integrity of data on the database, it’s make sure that the data is not corrupt as a result of failure, and it’s also guaranty about the validity of the data even when the failures occur.

Atomicity, it’s mean that all the data that has been modified are performed or neither of them will be. If one part of the transaction fails, all the transactions will fail. This provides Reliability.

Consistency, this make sure that the transaction maintains data integrity constrains and leaving the data consistent. The data that we saved on the database must be valid, from this way all illegal transaction will be avoid.

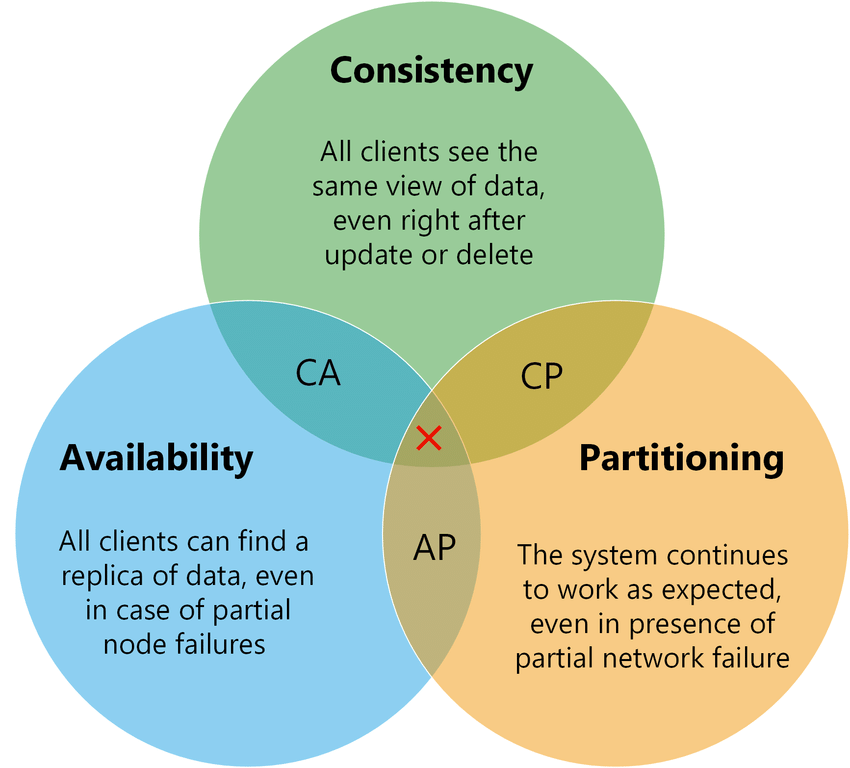
Isolation, this property make sure that a transaction can’t be change by other concurrent transaction until it’s completed.

Durability, this property means that the changes of the information that we made will be saved permanently until another update or delete transaction affects it.

* Relational database are table based on the roe and column and it’s mandatory to follow the standard of schema definitions. This is better for some application that need multi row transactions.
* On relational database it’s really difficult to do the horizontal scaling due to the flexibility of the query syntax. And why relational database is vertically scalable because they simply increase the power of the database server for e.g. by upgrading the CPU.

Non-Relational Database:

* the non-relational database is follow Brew’s cap Theorem is refers to Consistency, Availability, and Partition tolerance.

Consistency, this property is indicate that the read operation is getting the last record and it’s guarantee that all the information is UpToDate.

Availability, this property is indicate that the distributed system will be available even though one node will might turn off but the system still can access through another nodes.

Partition tolerance, it’s represent the ability of the system to partition, which is mean that each node can work independently from the others.

* Non-relational database have 4 type of storing, which is document data store, columnar data store, key-value store, graph databases.

Document data store, they manage to store the data in a document type such a JSON file, YAML file, XML file, or plain txt file, the document is not required to maintain the identical data structures, which is provide the flexibility.

Columnar data store, this method of storing the data is similar with the relational database, the real advantage of columnar data store is to denormalized approach for structuring the sparse data.

Key-value store, the key-value store is a collection of key-value pairs contained within an object.

Graph database, this method is made to efficiently store relations between entities.

* Non-relational database is horizontally scalable, they can handle the increased traffic by adding more server to the database. They have the ability to become larger and more powerful that’s what make them be choose for a large or constantly evolving the data set.

## When to use relational database

* Size of reputable data footprint - If you have a large amount of data that have repeating information, you can save space using a relational database format. Relational database Normalization can save a great deal of space by reducing or eliminating data redundancy.
* Data relationship complexity - How complex are the relationships inherent in your data. For example, when you write a query to a relational database how many complex relationships do you have to include in the query. If the answer is many, then database users have to spend more time writing these queries. This increases both development and operational time and costs.
* Predictable data- If your application is going to store predictable, structured data with a finite number of individuals or applications accessing it, a relational database is still a good option to consider

## When to use non-relational database

* Large amounts of data – NoSQL databases ensure data doesn’t become the bottleneck when all of the other components of your server-side application are designed to be seamless and fast.
* Little or no structure - NoSQL databases are either schema-free or have relaxed schemas.Do not require any sort of definition of the schema of the data
* Scalability and flexibility - The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data. The system response time becomes slow when you use RDBMS for massive volumes of data. To resolve this problem, we could “scale up” our systems by upgrading our existing hardware. This process is expensive.The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as “scaling out.”

# Conclusions

In conclusion both relational and non-relational offer their advanteages and disadvanteages. If you want : elimination of redundancy, complex relationship between and consistency, use data use relational database. If you want : fast writing speed , large and complex data use non-relational database. For my personal web shopping project, storing my data and the relations between it (e.g., User / Order / Item) a relational database is more suited for my goals and needs.

# References

*Which databases have you used in the last 12 months?* (n.d.). JET BRAINS. Retrieved

January 13, 2022, from <https://www.jetbrains.com/lp/devecosystem-2020/databases/>

Domareski, H. (2020, August 27). *ACID Properties*. Medium. Retrieved January 13, 2022,

from <https://henriquesd.medium.com/acid-properties-43e146b21e0d>

*SQL Vs NoSQL Exact Differences And Know When To Use NoSQL And SQL*. (2022, January

5). SoftwareTestingHelp. Retrieved January 13, 2022, from

<https://www.softwaretestinghelp.com/sql-vs-nosql/>

*Brewer’s CAP Theorem*. (2021, March 5). Baledug. Retrieved January 13, 2022, from

<https://www.baeldung.com/cs/brewers-cap-theorem>

Pattionson, T. (2021, August 13). *Relational vs. non-relational databases*. PLURALSIGHT.

Retrieved January 13, 2022, from

<https://www.pluralsight.com/blog/softwaredevelopment/relational-vs-non-relational-databases>

Schwarzmüller, M. (2018, July 25). *SQL vs NoSQL*. Academind. Retrieved January 13, 2022,

from <https://academind.com/tutorials/sql-vs-nosql>

*SQL, NoSQL, and Scale: How DynamoDB scales where relational databases don’t*. (2020,

January 6). Alexdebrie. Retrieved January 13, 2022, from

<https://www.alexdebrie.com/posts/dynamodb-no-bad-queries/>

Shiff, L., & Rowe, W. (2018, March 5). *SQL vs NoSQL Databases: What’s The Difference?*

BMC Blogs. Retrieved January 13, 2022, from <https://www.bmc.com/blogs/sql-vs-nosql/>

WhiteSource Software. (2021, July 29). *When to Consider a NoSQL vs Relational Database*.

WhiteSource. Retrieved January 13, 2022, from

<https://www.whitesourcesoftware.com/resources/blog/when-to-consider-a-nosql-vs-relational-database/>