**Research Documents (Database)**

**S3-CB04**

**Group 3 - Z. A. P**



**Group Members:**

Benjamin Brown (474338)[-b.brown@student.fontys.nl](mailto:-b.brown@student.fontys.nl)

Hristo Hristov (457108)[-h.hristov-ab@student.fontys.nl](mailto:-h.hristov-ab@student.fontys.nl)

Shanessa Kostaman (450080)[-genoveva.shanessakostaman@student.fontys.nl](mailto:-genoveva.shanessakostaman@student.fontys.nl)

Teun Visser (442651)[-teun.visser@student.fontys.nl](mailto:-teun.visser@student.fontys.nl)

Zep Alsters (463835)[-z.alsters@student.fontys.nl](mailto:-z.alsters@student.fontys.nl)

**Teacher:**

Roopali Gupta - [r.gupta@fontys.nl](mailto:r.gupta@fontys.nl)

**Client: Crossyn**

Bram van Herwijnen - [bram.van.Herwijnen@crossyn.com](mailto:bram.van.Herwijnen@crossyn.com)

**Fontys University of Applied Sciences**

**Eindhoven – Netherland**

**2020-2021**

Table of Contents

[Main Questions 3](#_Toc88051717)

[Sub Questions 3](#_Toc88051718)

[1. What’s the difference between relational database and non-relational database? 3](#_Toc88051719)

[2. When to use a non-relational database? 4](#_Toc88051720)

[3. When to use a relational database? 5](#_Toc88051721)

[Conclusions 6](#_Toc88051722)

# Main Questions

1. Which database is more suitable for the application?

## Sub Questions

### What’s the difference between relational database and non-relational database?

From the research that we did for the database, there’s a few difference between relational database and non-relational database.

Based on Terminology difference, SQL (relational database) store the tables in the database the table also consists of columns and they consist of rows/records . This is how the flow works :

***Database → table → column → records/rows.***

And NoSQL (non-relational database) store their collections in the database, then the collections will be a place to store the document (can be compiled into JSON/XML objects), and the document will store the key/field and value pairs (similar to a table containing only two columns). This is how the flow works :

***Collection →document*→ *key*and *value*pairs .**

Based on Structure difference SQL is a structured database and NoSQL is a unstructured database

### When to use a non-relational database?

1. When we need data storage that is "relatively" large , but there are limited resources (computers/servers). The relational database is actually scalable for big data even bigger from Petabyte scale. However, it takes a very large amount of resources to make a relational database perform well when the data is large.
2. When we have unstructured data (e.g. Documents) or commonly referred to as **Schemaless Data Representation** . For e. g. we have some column that can’t be null and we didn’t put any variable into that table. In relational database events like this will force us to change the DB structure (schema) which is usually very fatal, especially if the data is large. Schema changes are catastrophic for IT systems. Because the application must be revised following these changes. In Big Data terminology, this is related to “ Variety ”, which is the increasing variety of types/structures/formats of data that enter the database.
3. NoSQL fits perfectly into object-oriented applications .
4. With no Joins in NoSQL and complicated queries, NoSQL is famous for speeding up developers in building applications.
5. when the application / system requires a very large number of Write / Insert processes and in a short time. Associated with terminology in Big Data, NoSQL is very suitable if our system has a large data velocity .
6. When our data contains location information (e.g. Latitude/Longitude).
7. When we have Data complexity – data is stored (& managed) spread across different locations (data centres) (distributed).

### When to use a relational database?

1. When the inputted data has a very valuable value (e.g. payment/money transactions). NoSQL is really fast at inputting the data but the trade-off is not as reliable as other database types.
2. Analytic : despite having its own map reduce implementation, NoSQL is not a replacement for Hadoop or any other analytic engine. It is complementary (helpful) analytic engine.
3. Multi-Object Transactions : NoSQL does not support the input of multiple items into one or more “tables”. For e.g. MongoDB only supports for save the data into a document.
4. Complex Query : If the application that is built requires complex queries, NoSQL is usually not suitable. Although NoSQL has its own queries, it is more limited than conventional databases.
5. The 'nature' of the data is indeed relational (A==>nB, B==>nA).

# Conclusions

Based from our research we decide to use non-relational database (NoSQL), because we have to store a lot of data that we need to write to the database and most of the data is not related so we don’t have to connect one data table to another.