DAF Maintenance Prediction

Design Oriented Research: Report

Group 1 (Alex, Angel, Atanas, Stoyan, Tsanko)

# Introduction

This detailed research paper covers all the important questions we need to answer in order to reach our project goals. To tackle these questions, we're using the Development Oriented Triangulation Framework (DOT) framework and customizing methods for each smaller question. This way, we can carefully and methodically study the topic, which helps us gain a solid understanding and finish the project successfully. Down below, an overview of research questions will be given, after which a chapter with the initial research and outcomes on the corresponding sub-questions will be written. In the end, a conclusion with future recommendations will be mentioned.

# Research Questions and Methods

## Main question

How can we optimize the data visualization techniques currently used by DAF to detect potential component failures in the DAF engine plant?

## Sub-questions

* What does the current database schema look like? What are its drawbacks?
  + [Workshop (Root Cause Analysis)](https://ictresearchmethods.nl/Root_cause_analysis)

Understanding what the problem is and specifying what the focus of the solution should be.

* + [Field (Task Analysis)](http://ictresearchmethods.nl/Task_analysis)

Gain a proper understanding of the function of the database in relation to the project and the current design.

* + [Field (Problem Analysis)](https://ictresearchmethods.nl/Problem_analysis)

Making sure the problem definition is correct and that we are not solving the wrong problem.

* Is there a more optimal solution for data representation than Grafana?
  + [Library (Available Product Analysis)](https://ictresearchmethods.nl/Available_product_analysis)
  + [Library (Competitive Analysis)](https://ictresearchmethods.nl/Community_research)

A research document will be made containing all available Grafana alternatives and their pros and cons. In conclusion, the best-fitting visualization tool for this project will be highlighted.

* How can visualization usability be improved and made easy to use?
  + [Library (Community research)](https://ictresearchmethods.nl/Community_research)

Check out if others have had a similar problem and what they have done to resolve it. Any information gathered will be written down in a research document.

* + [Workshop (Prototype)](https://ictresearchmethods.nl/Prototyping)

An example graph displaying the temperatures of a machine will be made to go through the process of creating a query in order to observe the challenges.

* + [Lab (Usability testing)](https://ictresearchmethods.nl/Usability_testing)

A manual usability test will be performed and results will be laid out in a document.

* How can we expand the database to handle new sensor data?
  + [Lab (Data Analytics)](https://ictresearchmethods.nl/Data_analytics)

Gain an understanding of the needed data.

* + [Workshop (Prototype)](https://ictresearchmethods.nl/Prototyping)

Create diagrams and prototypes for the database.

* How can data be queried to ensure load times stay as low as possible?
  + [Lab (Non-functional test)](https://ictresearchmethods.nl/Non-functional_test)

General system validation on if it fulfills non-functional requirements agreed upon with the client in terms of performance.

* How can query complexity be minimized?
  + [Library (Design Pattern Research)](https://ictresearchmethods.nl/Design_pattern_research)

Investigation of the current approach for querying data and implementing on it the most recent design patterns for minimizing complexity.

* How can we ensure and optimize scalability for additional machines?
  + [Lab (Non-functional test)](https://ictresearchmethods.nl/Non-functional_test)

General system validation on if it fulfills non-functional requirements agreed upon with the client in terms of scalability.

# Research Analysis and Outcomes

## SQ-1: What does the current database schema look like? What are its drawbacks?

* Old ERD and drawbacks of its implementation and structure

## 

With the current implementation we have multiple problems. First everything goes through bfc\_source so there is not a lot of flexibility. The second problem is that all of the axes are grouped in tables of two. As a result you end up pulling a lot of unneeded data if you want only one of the axes.

## SQ-2: Is there a more optimal solution for data representation than Grafana and MySQL?

We looked into Grafana and database alternatives, to find a more optimal solution to store and represent the data. As for data representation, Grafana is still the best choice for this project, as it has every functionality needed for the scope and the clients have experience with it.

For database alternatives, we looked into many different time series and NoSQL solutions and the best ones were: **InfluxDB**, **CrateDB,** and **QuestDB**. The research methods used were *usability testing, literature study, and data analysis*.

We wrote a complete research file, which you can find in the project repository [here](https://git.fhict.nl/I450851/daf-maintenance-prediction/-/blob/main/docs/db-alternative-research.md). It contains the pros and cons of using databases other than MySQL and the steps (with instructions) to migrate the data. We've selected **QuestDB** as the most fitting database alternative for the project. This decision was influenced by several pivotal factors:

- **Full SQL Support**: QuestDB's encompassing support for SQL, including joins, guarantees a seamless transition from our current setup and eases the migration process.

- **Superior Performance**: Thanks to QuestDB's innovative columnar storage model and cutting-edge indexing algorithm, designed specifically for time-series data, we can expect a significant increase in data ingestion and querying speed. This feature is especially valuable for real-time analytics needs.

- **Exceptional Efficiency**: QuestDB's superb data storage efficiency and compression minimize resource usage in terms of memory and CPU, thus reducing the total cost of ownership.

- **Smooth Migration**: Shifting from MySQL to QuestDB is a relatively uncomplicated process, primarily due to QuestDB's strong SQL support. As we've demonstrated in our research, this procedure is both efficient and reliable, ensuring the integrity and consistency of our data.

Despite InfluxDB's established reputation and ecosystem, QuestDB surpasses it in key areas crucial to our project - SQL support, performance, and efficiency. Hence, in pursuit of enhancing visualization usability and overall project success, we've chosen **QuestDB** as our preferred database solution.

## SQ-3: How can visualization usability be improved and made easy to use?

Grafana Charts Normalization file: [link](https://git.fhict.nl/I450851/daf-maintenance-prediction/-/blob/main/docs/Grafana%20Chart%20Normalization%20-%20Research.pdf)

The document explains the main options Grafana provides for normalizing data and gives some examples of how different types of data could be visualized and put into common units. Furthermore, the document includes sequel steps in the sense of instructions on how to set specific chart normalization settings through the Grafana dashboard.

## SQ-4: How can we expand the database to handle new sensor data?

Expansion MD file [link](https://git.fhict.nl/I450851/daf-maintenance-prediction/-/blob/main/DafExpansion/alarmQueries.md):

In its current state, the DAF database has a problem when it comes to scalability. At this point, the main part of the DAF database will not be changed. However, the expansion needs to account for the potential discussed change and addition of new sensors while creating an expansion for new functionality concerning the alert system for the machines. For this, we first started with Filed, Document analysis to understand the current state of the database and how it works. Next with Workshop, Brainstorm to start thinking about how the expansion can be implemented while being guided by Library, Best, Good, and Bad practices. From there a Workshop, Prototype to be able to start testing. With the help of the team Showroom, Pear review we were able to find potential problems with the first prototype so an updated version was created and we Showroom, Pitch to the client. From that, we got a bit more clarification which led to the third version of the prototype. We confirmed with the client the final version of the expansion. With this, we have a DB expansion that works well for the alert system with the current database but it also accounts for future potential expansions.

## 

## SQ-5: How can data be queried to ensure load times stay as low as possible?

**Database optimization research**

The purpose of this research is to look into database optimization with the purpose of creating a better database structure and better querying in order to serve as a quality assurance guideline for the products of our group project. This research will be done using DAF’s current database structure however it will also focus on how it can potentially be improved in the future.

Database optimization research document: [link](https://git.fhict.nl/I450851/daf-maintenance-prediction/-/blob/main/docs/Database%20Optimization-research.pdf)

**Database Partitioning (or sharding)**

Database partitioning is a common tactic in database management, employed with the aim of breaking down large databases and tables into more digestible fragments known as partitions. By doing so, the fundamental objective of partitioning is to effectively enhance the overall performance, scalability, and maintainability of databases. This report contains information about database partitioning, exploring its advantages and showcasing its significance in the field of database management.

Database Partitioning document : [link](https://git.fhict.nl/I450851/daf-maintenance-prediction/-/blob/main/database-partitioning/partitioning.md)

## SQ-6: How can queries’ complexity be minimized?

**Summary tables and Event updating research**

MySQL summary tables are precomputed tables that store aggregated data derived from the original data in a database. They are used to enhance query performance by reducing the need for complex calculations during query execution. This report explores the concept of summary tables and discusses various methods to continuously update them in MySQL, including MySQL insert triggers, MySQL event schedulers, and other related techniques.

Summary tables and Event updating research: [link](https://git.fhict.nl/I450851/daf-maintenance-prediction/-/blob/main/summary-tables/summary_tables.md)

## SQ-7: How can we ensure and optimize scalability for additional machines?

As we started to Filed, interview the client it was quickly shown that this will be outside of the project's scope. This does not mean we completely disregarded it. As we were working on the other parts of the project we continuously kept a focus on the way this can change in the future and how new machines can be added. With the expansion of the database, we account for exactly that. No matter the way they add new machines our work can handle it.

# Conclusion and Recommendations

With all this, we can conclude that there was a lot of work that was done to create an overall good base for performance in the database. We were able to take this base and expand upon it with multiple tools. With this, we were able to improve the performance of the database but also to make the system more user-friendly.

After the work we have done, there are a couple of recommendations we can make. The current state of the database with how the axes are stored could be very problematic and it would be a good idea to change at the first available opportunity. Next, it would be advised to change to a different database or at least a different engine. With the testing and research we have done we can see that the MySQL team has done many improvements to the database however there still are problems that occur when it needs to work with a large dataset.