Technische Universität München

TUM School of Engineering and Design

Lehrstuhl für Computergestützte Modellierung und Simulation

Research Data Management and Reasoning on Indoor Climate Investigations

Bachelorthesis

für den Bachelor of Science Studiengang Umweltingenieurwesen

Autor:	Felipe de Albuquerque Sa Schuchmann
Matrikelnummer:	[redacted]
Prüfer:	Prof. Dr.-Ing André Borrmann
1. Betreuer:	M. Sc. Sebastian Esser
2. Betreuer:	M. Sc. Bilge Kobas
Ausgabedatum:	15. Oktober 2023
Abgabedatum:	15. Februar 2024

Foreword

In the continually changing world of research, effective data management and analysis are essential for driving innovation and discovery. As experimental methods grow more intricate, the demand for flexible Database Management Systems becomes critical. This thesis explores developing and implementing a hybrid Database Management System tailored to SenseLab's experimental needs.

SenseLab's diverse experiments provide an ideal testing ground for evaluating the hybrid Database Management System. By combining relational and graph-based databases, the system aims to tackle the unique challenges of SenseLab's varied datasets.

Throughout this thesis, the structure of the achieved hybrid Database Management System is explored, as well as the design process involved in developing this system. Furthermore, a practical analysis, through Case Studies and query scenarios, depicts the system's functionality, strengths, and limitations.

Abstract

This thesis explores the development and implementation of a hybrid Data Management System tailored to the data landscape of SenseLab's experimental framework. By blending relational and graph-based databases, the system aims to manage the diverse datasets generated by SenseLab's experiments effectively.

The research begins by analyzing SenseLab's data, followed by selecting criteria for the intended Database Management system, including data integration, management and future-proofness. This leads to the adoption of a hybrid Database Management System architecture.

Developing the Database Management System consists of different design stages. After implementing a basic structure, the research continues by testing the developed system with different case studies and query scenarios to evaluate the system's functionality. This includes cross-domain analyses and behavioral data exploration.

Following this stage, the benefits and limitations of the chosen architecture are made clear. Implementing the hybrid Database Management System faces software dependencies and multi-user access limitations.

The hybrid Database Management System offers a robust framework for managing the diverse dataset provided by SenseLab. By integrating relational and non-relational databases, the system demonstrates its capability to handle various data types and facilitate cross-domain analyses.

Summary

This thesis explores the development and implementation of a hybrid Database Management System tailored to meet the diverse data requirements of SenseLab's experimental framework. By blending relational and graph-based databases, the system addresses the challenges of heterogeneous datasets, offering a comprehensive solution for managing and analyzing experimental data. Through case studies and query scenarios, the thesis evaluates the functionality, strengths, and limitations of the hybrid Database Management System, providing valuable insights into its practical implications for experimental science.

Table of Contents

Table of Figures

Table of Tables

Abbreviations

DB	Database
DMS	Database Management System
ER-Diagram	Entity Relationship Diagram
GraphDMS	Graph Database Management System
NonRDMS	Non-Relational Database Management System
PMV	Predicted Mean Vote
RDMS	Relational Database Management System
RRDtool	Round-Robin Database tool
SQL	Structured Query Language

1 Introduction and Motivation

1.1 Research Questions

SenseLab is a climate chamber located on the main TUM Campus, where various experiments have been and are currently being conducted to redefine the concept of comfort and well-being in buildings. This is achieved by focusing on the human body's response to environmental conditions, emphasizing health more than previous research projects. [1]

Extensive time series-based data is being generated alongside these experiments, containing information about the climate inside the chamber, the behavior of each participant, and their physiological feedback and reactions. Each dataset has a different native resolution.

The amount of generated data is becoming a growing problem as the size and variety of data increase with each phase of the experiment. Therefore, a comprehensive data management system is required to meet the prerequisites and efficiently store, integrate, and visualize the various types of input data.

By adopting a combination of different DB, SenseLab not only ensures the efficient organization and storage of its heterogeneous datasets but also improves its data analysis capabilities. The selected DMS combination becomes a useful tool that empowers users, through various queries and scenarios to extract different information from the data. This enhances the analysis process and enables deeper exploration of the complex relationships between climate conditions, human behavior, and physiological responses. In essence, the chosen DB combination unlocks the full potential of the collected data.

This bachelor’s thesis aims to identify a DB combination that facilitates the analysis, interpretation, and retrieval of data collected by SenseLab. By using different database solutions, the aim is to strengthen SenseLab's capability to redefine comfort and well-being in buildings through efficient data management and analysis.

More specifically,

RQ1: What are the specific requirements and characteristics of the SenseLab data?

RQ2: What combination of relational and graph-based databases can combine the sequential and the topological nature of time-series and behavioral data?

RQ3: What is the most effective approach for integrating selected DMS solutions into a unified hybrid data management system for SenseLab?

To effectively address this research questions, several steps must be undertaken.

1. Comprehensive Data Understanding and Research: The initial phase of this endeavor involves a thorough study and in-depth comprehension of the data provided by SenseLab. Additionally, research will be conducted to explore various types of DMSs available today.

2. Evaluation and Selection of DMS Types: Building upon the knowledge acquired in the first step, the next phase of the thesis centers on evaluating and comparing different types of DMSs, taking the unique requirements of the SenseLab data into consideration. The goal is to identify which combination of DMSs is best suited to effectively manage the heterogeneous SenseLab datasets.

3. Integration of DMS Solutions: After the selection of suitable DMSs, the focus shifts to integrating these diverse systems into a unified platform. This integration process should not only ensure the smooth operation of the hybrid data management system but also be designed with scalability in mind.

4. Proposal of a Comprehensive Solution: As the culmination of this research and implementation effort, the thesis will draw conclusions. These conclusions will be synthesized into a proposal that outlines the hybrid DMS for SenseLab, providing a roadmap for efficient data handling and integration.

1.2 Structure of Thesis

Related works: Explores relevant databases and introduces SenseLab, providing a backdrop for understanding existing data management challenges and innovations in similar domains.

Methodology: Outlines the criteria for selecting a DMS for SenseLab, categorizing requirements, and evaluating the characteristics of relational and non-relational DMS types.

Development of a data management system for SenseLab: Details the design process of the hybrid DMS, incorporating both relational and non-relational components, and showcases the structure through an ER-Diagram and Cypher programming.

Case Study: This case study engages in a discussion of the benefits and challenges of the hybrid DMS, presenting two query scenarios that demonstrate its practical applications and effectiveness within the context of SenseLab.

Limitations: It depicts the different limitations that the developed DMS may face in the short- and long-term future, highlighting the different approaches to dealing with the mentioned limitations.

Summary: Finally, the summary presents a culmination of all the gathered information from the different chapters of this thesis. Showcasing the different approaches and conclusions drawn from this research.

2 Related works

2.1 ASHRAE

As described on ASHRAEs website, its mission is “to serve humanity by advancing the arts and sciences of heating, ventilation, air conditioning, refrigeration, and their allied fields” to achieve a “healthy and sustainable built environment for all”. [2]

The American Society of Heating, Refrigerating and Air-conditioning Engineers (ASHRAE) is known for developing and publishing standards and guidelines related to HVAC (Heating, Ventilation, and Air Conditioning) and refrigeration. [3]

2.1.1 ASHRAE Global Thermal Comfort Database

The tool “ASHRAE Global Thermal Comfort Database” was developed by ASHRAE to provide a collection of thermal comfort data from around the world and was the “first global scale attempt at thermal comfort data collection” [4]. This database is a valuable resource for researchers, designers, and practitioners in the field of building design and HVAC systems [5]. The human thermal comfort data stored within this DB includes environmental conditions, occupant characteristics and subjective feedback [5].

The first version of the ASHRAE DB was developed by Richard de Dear in 1998, and it included survey results from 52 unique buildings out of 160 surveyed between 1982 and 1997. The second version, ASHRAE II, which is still in use today, was released 20 years after its predecessor. [4]

The nowadays still in use ASHRAE Global Thermal Comfort Database II contains multiple field studies conducted between 1995 and 2016 from different parts of the world. Distinct data was collected during each of these studies, including Climate, Building, Demographic and Subjective data. The DB is equipped with an “Interactive thermal comfort data visualization tool” aiming to provide a user-friendly interface. This tool is subdivided into three pages: satisfaction scores, adaptive comfort and scatter plots. Bellow the output graph section, users can filter the data by four different categories, including 49 distinct parameters, enabling users to create different subsets and perform analysis queries best suited for their specific needs. [5]

2.1.2 Limitations

Despite its abilities, ASHRAE's Global Thermal Comfort Database II does not meet the standards required for properly assessing room comfort. As discussed in the paper "Towards a Multivariate Time-Series Approach with Bio signal Datasets for the Global Thermal Comfort Database" several improvements must be made to the mentioned database to fully achieve its intended purpose. These improvements include incorporating time data and physiological data, enhancing demographic data, and providing a better description of the sensors used during the experiments. [4]

Different NoSQL databases already meet diverse requirements, as depicted in the paper "Suitability of InfluxDB Database for IoT Applications" [6]. For instance, "Graphite" is a highly configurable real-time graphing system, and "RRDtool" is an open-source Round-Robin Database tool [6] specialized as a system designed for efficient storage and visualization of time-series data [7]. Developed to address the challenges of storing large amounts of temporal data while maintaining a constant database size, RRDtool utilizes a circular storage solution [7]. This solution functions by overwriting the oldest data with the most recent, allowing for continuous data recording while also controlling the DB expansion [7].

While updating ASHRAE's Global Thermal Comfort Database to its current version saw an increase in size, a clear lack of continuity regarding key structural features between the two versions was noticeable. These differences include the absence of time stamps and date data, as well as the lack of a clear identifier that was used in the first version to differentiate subjects from buildings. [4]

This deficit highlights a significant gap and underscores the need for the development of a third, more comprehensive, and robust version of this type of data management system. SenseLab has taken up this challenge and is generating rich data with each phase of its project.

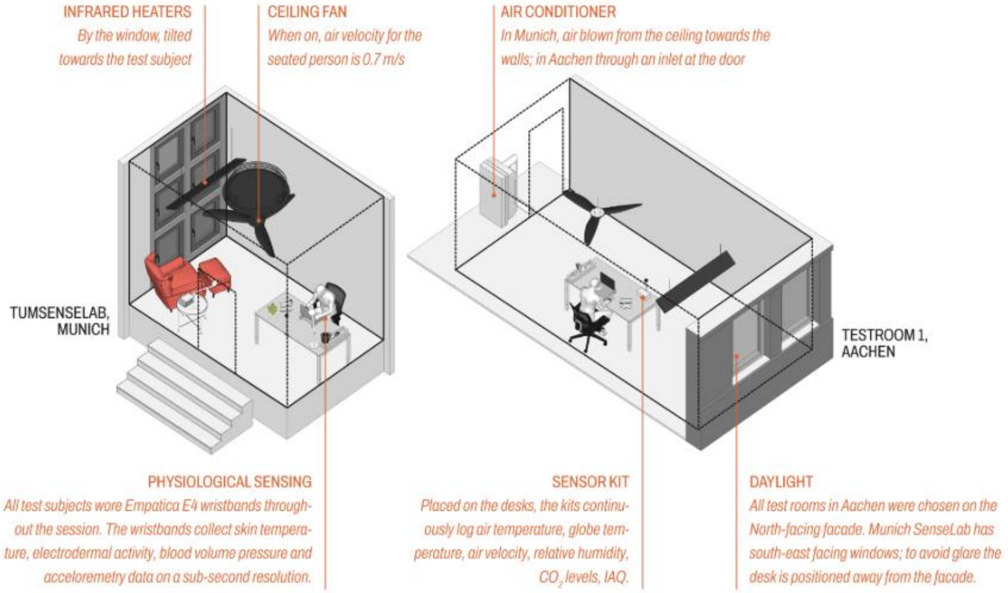
2.2 SenseLab

The SenseLab project seeks to revolutionize the understanding of comfort within built environments by adopting a unique perspective, examining the human body directly [1]. In response to longstanding uncertainties surrounding the numerical definition of comfort, the project recognizes that such definitions have far-reaching implications, impacting occupants' well-being. Addressing concerns about bias, data resolution, and scalability in comfort literature, SenseLab diverges from traditional approaches by focusing on the physiological markers within the human body. This multidisciplinary initiative involves collaboration among architects, building engineers, and researchers with medical backgrounds [1]. By delving into the intricate relationship between perceived comfort and physiological responses, SenseLab aspires not only to redefine comfort but also to gather long-term data highlighting the relationship between indoor environments, human health and well-being [8].

2.2.1 Experiment Setup and Participants

The experiment takes place in three repurposed office rooms in Aachen and a compact climate chamber constructed on the TUM main campus. Each of these spaces is outfitted with controllable climate systems and sensing equipment. [8]

Notably, none of the testing facilities possesses the capability to regulate the humidity. Considering this limitation, an average value was computed using values from previous experiments [8]. “Figure 1“ illustrates an experiment room in Aachen and the SenseLab in Munich.



INFRARED HEATERS
By the window, tilted towards the test subject

CEILING FAN
When on, air velocity for the seated person is 0.7 m/s

AIR CONDITIONER
In Munich, air blown from the ceiling towards the walls; in Aachen through an inlet at the door

PHYSIOLOGICAL SENSING
All test subjects wore Empatica E4 wristbands throughout the session. The wristbands collect skin temperature, electrodermal activity, blood volume pressure and accelerometry data on a sub-second resolution.

SENSOR KIT
Placed on the desks, the kits continuously log air temperature, globe temperature, air velocity, relative humidity, CO_2 levels, IAQ.

DAYLIGHT
All test rooms in Aachen were chosen on the North-facing facade. Munich SenseLab has south-east facing windows; to avoid glare the desk is positioned away from the facade.

Figure 1: Test Rooms in Aachen and Munich [8]

During the already concluded first phases of the experiment, four distinct scenarios were created. These scenarios were crafted to emulate real-life situations and yield an identical PMV despite featuring different individual parameter values. [8]

Table 1 description of the four different test scenarios [8]

Scenarios	**Description**	**Operative Temperature (top) (°C9)**	**Air Temperature (tbd) (°C)**	**Globe Temperature (tg) (°C)**	**Mean Radiant Temperature (tr) (°C)**	**Air velocity (v) (m/s)**	**Clothing factor (clo)**	**Relative humidity (rh) (%)**
Test 1	Baseline	27.19 (27.45)	27.17 (27.90)	27.17 (27.40)	27.20 (27.00)	0.1	0.48	43 (30)
Test 2	Same top as T1, asymmetry between tbd and tr	27.23 (27.25)	25.00 (25.00)	27.23 (27.40)	29.45 (29.50)	0.1	0.48	43 (34)
Test 3	Warmer + air movement	30.79 (31.05)	29.80 (30.00)	30.80 (31.00)	33.10 (33.50)	0.8	0.48	43 (34)
Test 4	Hot + air movement + low clo	32.19 (32.38)	31.20 (32.00)	32.20 (32.15)	34.50 (32.50)	0.8	0.24	43 (28)

The testing sessions were conducted daily, each spanning 6 hours, divided into two three-hour long sessions, separated by an hour-long lunch break. [8]

“Figure *2* “ represents a typical experiment day:

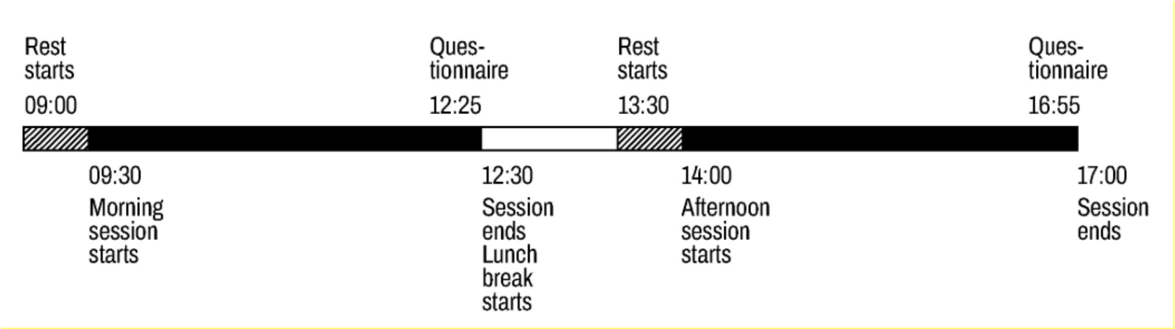


Figure 2: Typical experiment day[8]

Looking for a diverse participants group, no specific BMI or age restrictions were imposed upon the 22 participants. The sole stipulated conditions included restriction with individuals with active illnesses, ongoing medication use, pregnancy, irregular sleep-wake cycles, fluctuating body weight, smoking habits, or regular alcohol consumption.[8]

2.2.2 The Data

The data collected by SenseLab ensures a comprehensive dataset for analysis, covering both the environmental conditions and the physiological responses and behaviors of the participants. [8]

It can be divided into four categories: Climate data, Behavioral data, Physiological data, and Occupant feedback. Each category consists of specific aspects crucial for analysis.

“Figure *3*“ provides an overview of the sensor placement in the SenseLab climate chamber. “Table *2*” offers a description of the data collected by each sensor.[8]

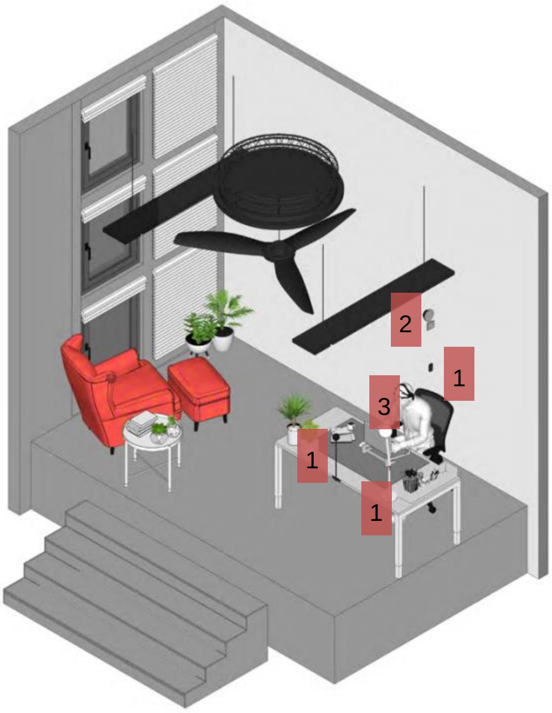


Figure 3: Position for the different sensing equipment [9]

Table 2 SenseLab data description [8]

Data Type	Data Description	Measurement Details	Resolution
1. Climate Data	Environmental conditions under which the experiment is conducted	- Air temperature (tdb) - Air movement (v) - Relative humidity (rH) - Indoor air quality (IAQ) - Co2 concentration	Once per minute
2. Behavioral Data	activities of the test volunteer during the experiment	- Break-Times - food/ drink intake -relevant comments - Clothing levels (clo: ASHRAE XX)	Once per minute
3. Physiological Data	Measurements	- Blood volume pressure (BVP) - Wrist skin temperature - Wrist electrodermal activity (EDA) - Motion	EDA: 4HZ Skin Temperature: 64Hz
4. Occupant feedback	Thermal satisfaction and preference questionaries	- Questionaries	Handed out 5 min before the end of each session

Additionally, metadata for each session and participant is systematically collected and stored, encompassing:

Table 3 Metadata SenseLab [8]

Metadata	Description
Participant	Self-reported information collected at the beginning of each session: - Participant ID - Age - Sex - Weight - Height
Sessions	- Start- and end time - Location (SenseLab or Aachen)

2.2.3 Sensing Equipment

Each data group described earlier employs distinct sensors for data gathering. Throughout the experiment, Climate and Physiological data are consistently recorded. The climate data is gathered using a custom sensor kit, while Physiological data is captured through Empatica E4 wristbands. [8]

In contrast, Behavioral and Feedback data are inputted manually by the observers. Researchers actively engage in the process of logging Behavioral observations and obtaining Feedback from participants. [8]

“Table *4*” provides an overview of the collected data.

Table 4 Information on sensors [8]

Data	Unit	Sensor Type	Brand	Sampling rate	Accuracy
Air temperature	°C				
Relative humidity	%				
Globe thermometer	°C				
Air velocity	m/s				
Air quality					
CO2 concentration					
Outdoor air temperature	°C				
Electrodermal activity (EDA)	µS	2 skin electrodes	Empatica	4 Hz	
Photoplethysmography (PPG)/ Blood volume pulse (BVP)	nanoWatt	PPG sensor	Empatica	64 Hz	
Skin temperature	°C	Infrared thermopile	Empatica	4 Hz	
Motion-based activity	g	3-axis-accelerometer	Empatica	32 Hz	

2.2.4 Preprocessing of recorded values

The recorded raw physiological signal underwent numerous processing stages to yield the desired dataset format. The electrodermal activity signal underwent a filtering and decomposition process using the neurokit2 Python package, yielding tonic and phasic signals. Skin temperature was filtered through a 1-minute moving window. The photoplethysmogram signal was processed using the “heartPy” Python package. This process involves the filtering, cleaning and computation of heart rate parameters.[8]

Following filtering, cleaning and feature extraction, all signals were down-sampled to a 30-second resolution using a Fourier method within the “sciPy.signal” package in Python. Following that, PMV values were calculated using the “pythermalcomfort” package. [8]

The processed signals were then timestamp-matched with climate data and behavioral observation and each session was masked by a start and an end time. [8]

To prepare the resultant time-series dataset for analysis, rows marked as breaks and those within 2 minutes of entering and leaving the room were excluded. This process ensured the creation of a reliable dataset, setting the stage for future analyses.[8]

2.3 Potential Database Management System

When assessing the data provided by SenseLab, there are several types of DMS that could be implemented to meet specific needs. Two primary categories include the RDMS and the varied array of NonRDMS, commonly known as NoSQL DMS. The second includes different types of DB, such as document-oriented, key-value stores, column-family stores and graph databases.[10] Each DMS within these categories has its unique characteristics and limitations, best suited to different data structures and use cases.

2.3.1 Relational Data Management Systems

The Relational Database Management System (RDMS) model was introduced by Edgar Codd in his paper "A Relational Model of Data for Large Shared Databank” in the 1970s. [11] Since then, it has played a pivotal role in data management.

It is a traditional and well-established type of DMS that organizes data into tables with defined relationships between them [12]. It is best suited for well-structured data with a predefined schema, making it effective for scenarios where data integrity is crucial, leaving no space for data redundancy. [12] Remarkably, it can handle vast amounts of information, scaling up to petabytes, as demonstrated by platforms like Twitter and Facebook. [13] Some examples of RDMSs are MySQL, PostgreSQL, and Oracle Database.

RDMS excels when dealing with well-structured tabular-like information.[12] It primarily utilizes the SQL language to manage the system. RDMS is well-regarded for its already mentioned reliability and efficiency, excelling in scenarios where the structured data exhibits straightforward relationships, making them the preferred solution for traditional data management tasks. [14] Although not specifically optimized for this, RDMS proves useful when dealing with time series data, showcasing its versatility and applicability across various data domains. [15]

It complies with the ACID (Atomicity, Consistency, Isolation and Durability) format. These “are a set of governing principles of the relational model” [14] meant to guarantee DB reliability.

However, challenges arise when dealing with data rich in complex relationships, necessitating numerous joins to process queries. In the realm of RDMSs, the complexity of relationships can pose significant hurdles. As datasets grow in size and intricateness, the need for multiple joins becomes more pronounced. [14]

In RDMS, joins are operations that combine rows from two or more tables based on related columns [16]. While this structure offers a flexible and efficient way to manage interconnected data, it introduces challenges when querying information that spans multiple tables [16]. This can result in slower response times for queries and a potential strain on the overall DB performance.

In essence, while RDMS provide an effective way of organizing and relating data, the management of complex relationships introduces challenges that necessitate thoughtful optimization strategies for maintaining optimal performance in the face of evolving and interconnected datasets.

Despite these challenges, RDMS remains the predominant choice for data management and is widely utilized.[14]

Some examples of possible server-based RDMS are [12], [11]:

Table 5 Possible RDMS

Relational DMS:	Description:
MySQL	Open source Commonly used in Web-applications Known for its speed and reliability
Microsoft SQL server	Comprehensive RDMS Commonly used in Windows environments
PostgreSQL	Open source Known for its extensibility. Support for advanced data

2.3.2 Non-Relational and NoSQL Data Management Systems

An emerging alternative gaining momentum is the Non-Relational DMS or NoSQL DMS. This surge can be attributed to the limitations experienced by RDMS when dealing with datasets characterized by intricate relationships.

As described in the paper “A Comparison of a Graph Database and a Relational Database”, some indications that NoSQL DB is the better option are [14]:

- Having data with Tables with lots of poorly used columns,
- Having attribute tables,
- Having lots of many-to-many connections,
- Having tree-like characteristics and
- Requiring frequent schema changes.

NoSQL DMS presents a diverse range of options, each designed to handle specific data structures and requirements. [17]

- **Document-Oriented Databases** [17]**:**

Document-oriented databases like MongoDB store data in flexible, JSON-like documents, making them suitable for semi-structured or unstructured data. These systems are advantageous for applications that require flexibility in data representation.[17]

- **Key-Value Stores** [17]**:**

Key-value stores, exemplified by systems like Redis and Amazon DynamoDB, are highly efficient for simple data retrieval using unique keys. They excel in scenarios requiring fast and scalable data access, prioritizing scalability over consistency.[17]

- **Column-Family Stores** [17]**:**

Column-family stores, such as Apache Cassandra, organize data into columns rather than rows, optimizing read and write performance. This makes them suitable for scenarios with high write and query loads.[17]

- **Graph Databases** [18]**:**

Graph databases, like Neo4j, focus on relationships between entities, making them ideal for scenarios where understanding and querying complex relationships are crucial. [18]

While there were earlier attempts to implement the already mentioned ACID properties in NoSQL databases through additional programming for compliance, the adoption of ACID has been limited in the NoSQL domain primarily due to performance considerations. [11]

Therefore, NoSQL databases do not adhere to the traditional ACID format. Instead, many NoSQL databases embrace the BASE model, standing for “Basically Available”, “Soft-state”, “Eventually consistent”. BASE provides an alternative set of principles that align better with the distributed and decentralized nature of NoSQL systems, favoring “availability, grace degradation and performance over ACIDs properties “Consistency” and “Isolation”. [17]

“Basically Available” emphasizes the importance of maintaining system availability even in the face of partial failures; in other words, an application works all the time, “Soft state” allows for temporary inconsistencies, and “Eventually consistent” ensures that the system will reach a consistent state over time. [17] [19]

In summary, this approach is well-suited for handling the diverse data formats often encountered in NoSQL databases, enabling efficient management of large volumes of data across environments without the strict constraints of ACID.

2.3.3 MySQL Relational Database Management System

MySQL is an open-source RDMS known for its use among websites and great speed and reliability. It relies on the SQL language to run queries. [12]

MySQL Workbench enables users to create and manage connections to DB servers within MySQL servers while enabling users to run SQL queries through its integrated SQL editor. It is a tool supported by MySQL servers beyond version 5.7.[20]

The MySQL Workbench greatly helps users unfamiliar with the SQL language when developing a DB, making MySQL server a good option for beginners.

2.3.4 Neo4j Graph Database Management System

The Neo4j GraphDMS is a fully ACID compliant, open-source DB. Nowadays, it is the most popular GraphDMS among the available options. [21]

Unlike other DMS, Neo4J offers a versatile approach to storing data rich in connections. Prioritizing the integrity of these relationships. It utilizes the GraphDMS querying language Cypher, which is one of the easiest graph query languages to learn, making it very simple for users to familiarize themselves with the DB. [22]

Through the APOC library, Neo4J gains the remarkable capability to seamlessly integrate various types of DMS and files into its schema, creating “virtual nodes” within the created graph for the data stored in other locations. [23]

Ultimately, Neo4j's adaptability and approach when dealing with data rich in connections not only facilitates the integration of different data sources but also positions it as a versatile solution for structuring this data.

2.4 Current State

ASHRAE’s DB uses MySQL backstage to store its data and process queries [5], but to best depict all the collected data by SenseLab, different approaches must be taken into consideration.

For instance, various researchers are using GraphDMS to store, manage, visualize and query their data. In the construction area, as depicted in the paper “Object Detection Based Knowledge Graph Creation: Enabling Insight into Construction Process”, using GraphDMS like Neo4J is especially useful when structuring data in a way that represents and preserves its process. [24] The paper discusses the development of a knowledge graph to store and connect data from construction sites, showing that GraphDMSs are effective for storing data in a way that its context doesn't get lost. [24]

Also, within the construction domain, as shown in the paper "Enriching Building Graphs with Parametric Design Constraints for Automated Design Adaptation", using GraphDMS when planning a building can be extremally useful, especially considering the GraphDMS ability to be easily updated during the design process [25], making it a great storage option for changing data.

Given the complexity, heterogeneity and general aspects of the data collected within SenseLab's context, fully achieving its goals involves incorporating its data into the developed DMS. A hybrid DMS in which both relational and graph DB approaches are used is the best-suited solution compared to using a single DB system.

3 Methodology

When considering the architecture of the DMS for SenseLab, the decision to incorporate both relational and graph-based databases is driven by the unique characteristics of the data and the diverse requirements of the experiments.

SenseLab generates extensive time series-based data containing information about climate conditions, participant behavior, and physiological feedback, each with its own distinct structure and relationships (2.2).

As previously depicted, RDMS excel in handling structured data with predefined relationships (Chapter: 2.3.1), making them suitable for aspects like climate and physiological data. However, RDMS may struggle to efficiently capture the complex connections encountered when analyzing the behavioral observations.

On the other hand, GraphDMS are optimized for storing and managing complex relationships and are well-suited for representing the intricate connections among the recorded behavioral observations. (Chapter: 2.3.2)Non-Relational and NoSQL Data Management Systems

By adopting a hybrid DMS approach, we aim to leverage the strengths of both relational and graph DB to provide a more complete solution for managing the heterogeneous data generated by SenseLab. This decision is grounded in the recognition that different types of data exhibit varying structures and relationships, and a hybrid DMS offers the flexibility needed to navigate this context.

3.1 Requirements

Careful consideration of SenseLab's specific requirements is paramount in aligning them with the objectives of the hybrid DMS when selecting criteria for the different data scenarios.

Based on the different DMS presented in Chapter 2.3 and on the SenseLab data presented in Chapter 2.2 it is beneficial to categorize the criteria into two groups: those impacting present data use and those considering future needs.

An essential consideration is data migration, evaluating the efficiency and quality of transferring previously collected data from the current ".csv" files to the new DMS.

Criteria influencing present data use encompass data integration, management/storage and collaboration among users.

In terms of data integration, assessing the DMS's support for the different data formats is crucial. For data management and storage, the DMS's ability to handle both structured and unstructured data is essential. Collaboration examines how effectively the system performs when accessed by different users simultaneously, an important consideration for the collaborative project between SenseLab and Aachen.

In the realm of future considerations, factors such as scalability and future-proofness must be considered.

Assessing scalability refers to the DMS's ability to expand both horizontally and vertically as data variety and quantity increase. Future-proofness is crucial, requiring the selection of a DMS well-supported by its founding company to ensure continuous updates and the addition of new features for years to come.

A more traditional approach when selecting the criteria needed to choose a DMS is analyzing the classic capabilities of a DMS. It should be able to edit, update, input, select, share, manipulate and display the stored data [26]. Although not named in the same way, these aspects can be found within the already mentioned criteria.

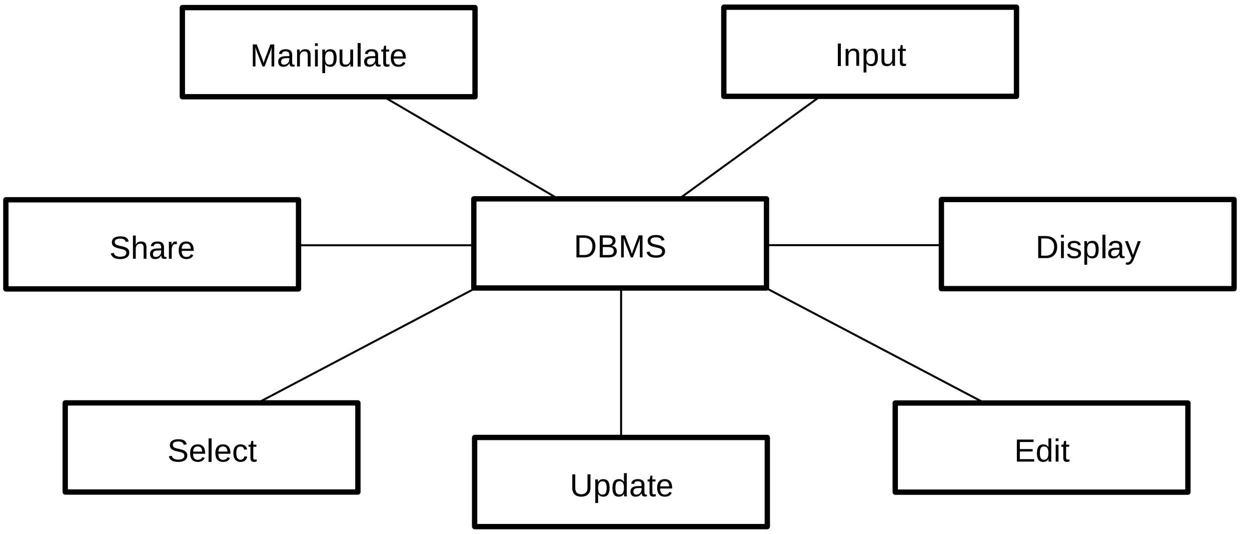


Figure 4: Capabilities of a DMS [26]

3.2 Evaluation of the DMS

To effectively determine the most suitable DMS for different types of data, careful consideration of proposed criteria is important.

Upon comparing the provided data with the characteristics of both types of DMS, it becomes evident that physiological and climate data are best managed by an RDMS. This choice aligns with the nature of these data types, which are inherently two-dimensional, well-structured data and can be optimally represented in a tabular format, as explained earlier (2.3.1).

On the other hand, when dealing with provided behavioral data, the consideration of a NonRDMS comes to the forefront. The collected data contains varying structures and is rich in connections; therefore, it does not fit neatly into a tabular format. A NonRDMS is better adapted for handling the flexibility required by diverse data structures in this context.

With these considerations in mind, it is relevant to analyze how each of the data categories would fit within the already discussed DMS in chapters 2.3.3 and 2.3.4.

MySQL Server, together with MySQL Workbench, offers a robust system for implementing the collected climate and physiological data (2.3.3). Importing, storing, managing and querying the data is facilitated through various features offered by MySQL Workbench [20], and considering the consecutive integration between the different DMS, MySQL can be easily accessed through Python using the “mysql-connector-python” driver, directly provided by Oracle [27]. MySQL is also constantly receiving updates and improvements, putting itself in a good position when considering future-proofness [20].

Neo4J is in a similar position, offering a suitable solution for importing, storing, managing, and querying the behavioral data stored in it (2.3.4). These different operations are facilitated by the simple and effective use of Cypher queries and the wide support from Neo4J for different file formats [22]. When considering the integration within the hybrid DMS, it is also in a similar position to MySQL. Utilizing the driver “neo4j” for Python allows the user to access the data easily through Python [28].

4 Development of a data management system for SenseLab

To embark on the experimental phases of this thesis, the initial step involves the meticulous design of each DMS, encompassing both relational and non-relational components. Employing an Entity Relationship Diagram emerges as the most convenient approach for developing the structure of the RDMS. This visual representation aids in conceptualizing the relationships, entities, and attributes, providing a strategic blueprint for organizing and managing the data in a structured manner.

Simultaneously, for the non-relational DMS, achieving optimal results involves the utilization of Cypher to program its layout directly into the Neo4j GraphDMS. This approach ensures that the NonRDMS aligns seamlessly with the inherent complexities of the provided data.

4.1 Relational Database Management System

In the context of the RDMS, it is important to develop a structure that best storages the climate and physiological data to achieve the intended objective.

4.1.1 Structure and Preprocessing

To properly represent the experiment, it’s important to first determine the required entities, their attributes, primary keys, and relationships. All the collected data is provided within the same “.csv” file, accompanied by designated timestamps.

After analyzing the provided data, it’s clear that to structure the RDMS, the following entities are needed: Sessions, Labs, Participants, as well as Climate and Physiological data.

To effectively integrate this data, it is imperative to preprocess it to meet our specific requirements. During preprocessing, it's important to extract the necessary information to structure each entity.

Regarding the “session” entity, essential details such as the start and end times, the scenario used, and the corresponding “session_id” must be extracted from the data. These attributes are crucial for accurately representing the different sessions within the RDMS. Notably, the “session_id” holds utmost significance as it serves as both the primary key for this entity and a foreign key within climate and physiological data entities.

As highlighted earlier, for both Climate and physiological data, the “session_id” plays a fundamental role and must be stored for each case. The “session_id”, once implemented, establishes an important relationship between each data point and its corresponding session. Besides the “session_id” and “timestamps”, no other measured data overlaps between these two entities.

For climate data, the preprocessing should focus on parameters such as CO2 concentration, relative humidity, air velocity, air temperature and internal air quality data, among others. In contrast, for physiological data, the preprocessing encompasses parameters such as motion, electrodermal activity, wrist skin temperature, and blood volume pulse measured values.

To accurately represent each session and the overall experimental structure, it is essential to account for the different Labs and Participants.

When considering the different “Labs”, two pertinent attributes are the “Lab_id” and Lab name, providing a clear distinction between the SenseLab and the three experiment Labs in Aachen. This distinction is vital for organizing and categorizing the data effectively. The “Lab_id” acts as the primary key for each lab.

Similarly, for participants, utilizing all recorded metadata is essential to differentiate between each one properly. Additionally, the BMI, BSA, and “Participant_id”, which acts as the primary key, for each participant are stored.

In summary, thorough preprocessing is required to ensure the integration and accurate representation of both Climate and physiological data, considering the unique attributes associated with each data type, session, lab, and participant.

In “Figure 5”, the depicted ER-Diagram illuminates the adopted structure for the RDMS.

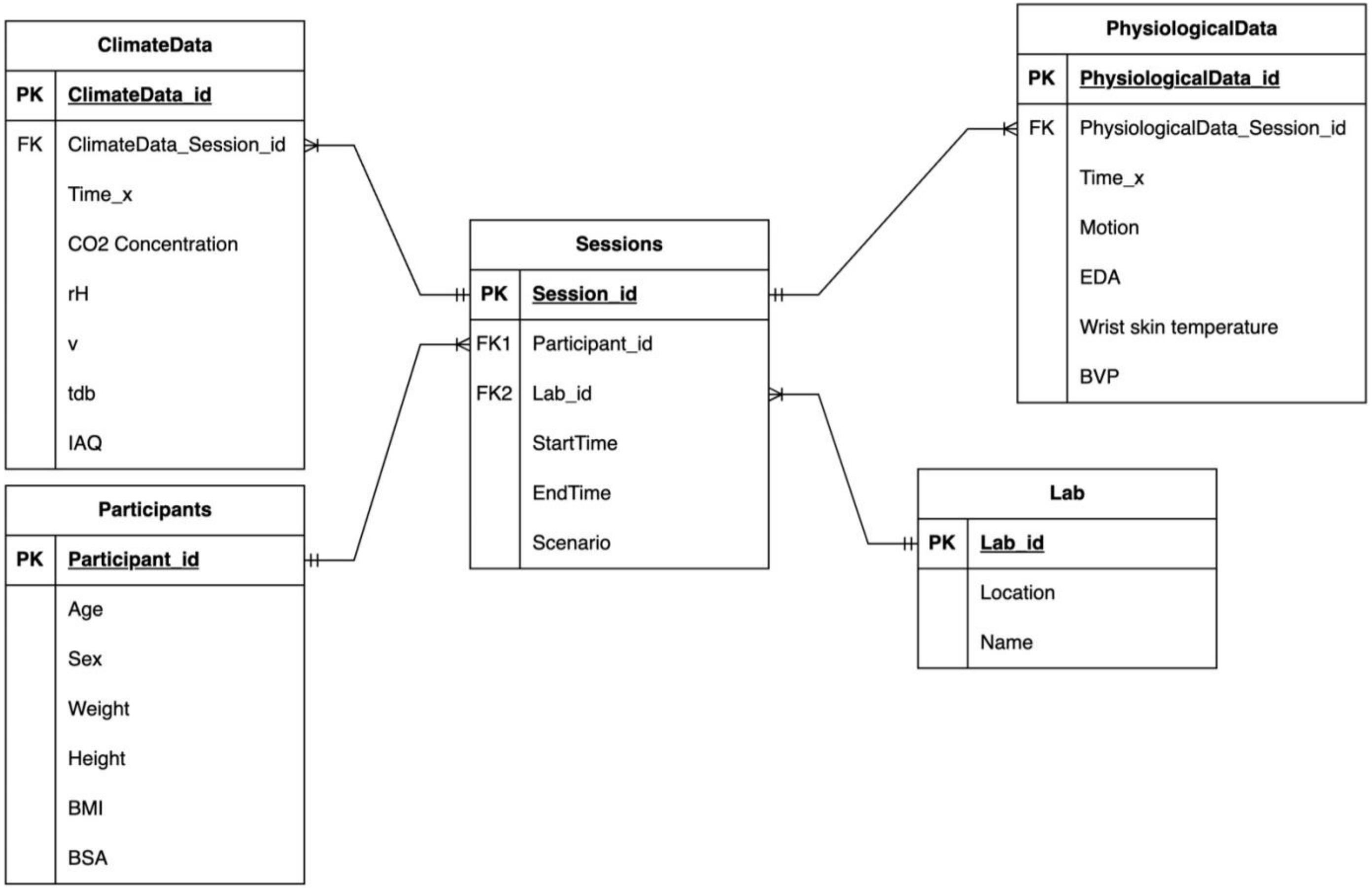
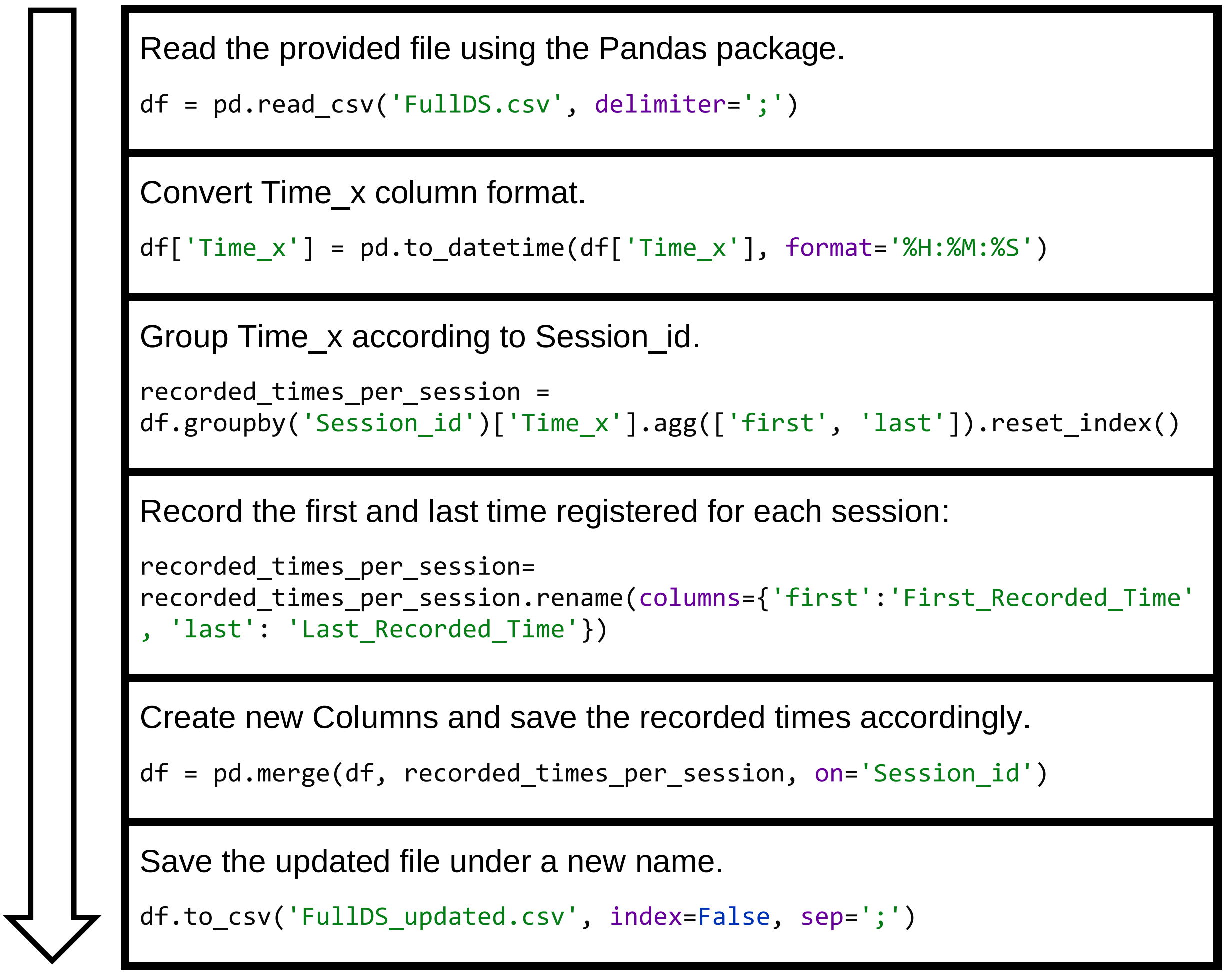


Figure 5: ER-Diagram for RDMS

The preprocessing is performed by different Python scripts designed to process the provided data and tailor it to the specific needs of SenseLab's project. For instance, the "Start&End.py" Python script was used to assign to each session the correct start and end times.

The following figure represents the code process:



Read the provided file using the Pandas package.

```
df = pd.read_csv('FullDS.csv', delimiter=';')
```

Convert Time_x column format.

```
df['Time_x'] = pd.to_datetime(df['Time_x'], format='%H:%M:%S')
```

Group Time_x according to Session_id.

```
recorded_times_per_session =
df.groupby('Session_id')['Time_x'].agg(['first', 'last']).reset_index()
```

Record the first and last time registered for each session:

```
recorded_times_per_session=
recorded_times_per_session.rename(columns={'first':'First_Recorded_Time'
, 'last': 'Last_Recorded_Time'})
```

Create new Columns and save the recorded times accordingly.

```
df = pd.merge(df, recorded_times_per_session, on='Session_id')
```

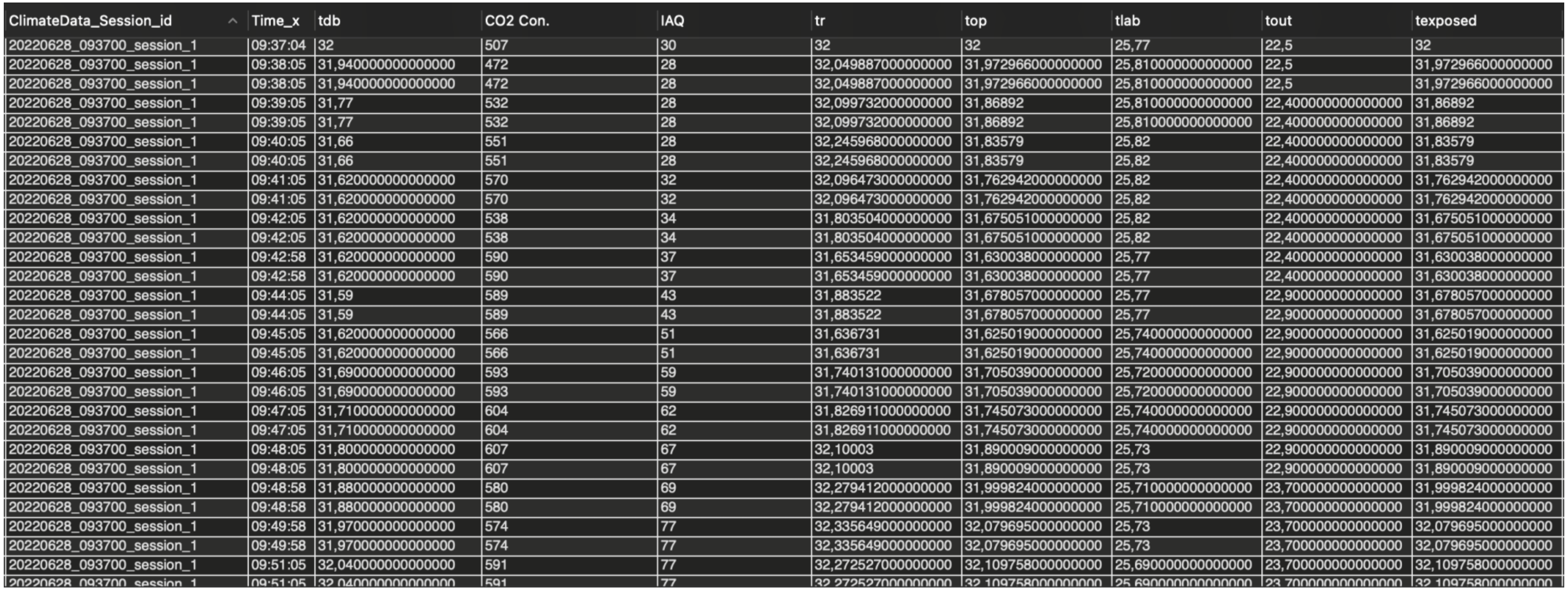
Save the updated file under a new name.

```
df.to_csv('FullDS_updated.csv', index=False, sep=';')
```

Figure 6: "Start&End.py" code Structure

After preprocessing the Data, it can easily be imported into the RDMS using MySQL Workbench. Through the “Table Data Import Wizard” function, the data can be easily assigned to the correct previously implemented attribute within the MySQL environment.

In “Figure 7”, a small section of the climate data is depicted. Physiological data is similarly structured, differentiated only by its own attributes.



ClimateData_Session_id	Time_x	tdb	CO2 Con.	IAQ	tr	top	tlab	tout	texposed
20220628_093700_session_1	09:37:04	32	507	30	32	32	25,77	22,5	32
20220628_093700_session_1	09:38:05	31,940000000000000	472	28	32,049887000000000	31,972966000000000	25,810000000000000	22,5	31,972966000000000
20220628_093700_session_1	09:38:05	31,940000000000000	472	28	32,049887000000000	31,972966000000000	25,810000000000000	22,5	31,972966000000000
20220628_093700_session_1	09:39:05	31,77	532	28	32,099732000000000	31,86892	25,810000000000000	22,400000000000000	31,86892
20220628_093700_session_1	09:39:05	31,77	532	28	32,099732000000000	31,86892	25,810000000000000	22,400000000000000	31,86892
20220628_093700_session_1	09:40:05	31,66	551	28	32,245968000000000	31,83579	25,82	22,400000000000000	31,83579
20220628_093700_session_1	09:40:05	31,66	551	28	32,245968000000000	31,83579	25,82	22,400000000000000	31,83579
20220628_093700_session_1	09:41:05	31,620000000000000	570	32	32,096473000000000	31,762942000000000	25,82	22,400000000000000	31,762942000000000
20220628_093700_session_1	09:41:05	31,620000000000000	570	32	32,096473000000000	31,762942000000000	25,82	22,400000000000000	31,762942000000000
20220628_093700_session_1	09:42:05	31,620000000000000	538	34	31,803504000000000	31,675051000000000	25,82	22,400000000000000	31,675051000000000
20220628_093700_session_1	09:42:05	31,620000000000000	538	34	31,803504000000000	31,675051000000000	25,82	22,400000000000000	31,675051000000000
20220628_093700_session_1	09:42:58	31,620000000000000	590	37	31,653459000000000	31,630038000000000	25,77	22,400000000000000	31,630038000000000
20220628_093700_session_1	09:42:58	31,620000000000000	590	37	31,653459000000000	31,630038000000000	25,77	22,400000000000000	31,630038000000000
20220628_093700_session_1	09:44:05	31,59	589	43	31,883522	31,678057000000000	25,77	22,900000000000000	31,678057000000000
20220628_093700_session_1	09:44:05	31,59	589	43	31,883522	31,678057000000000	25,77	22,900000000000000	31,678057000000000
20220628_093700_session_1	09:45:05	31,620000000000000	566	51	31,636731	31,625019000000000	25,740000000000000	22,900000000000000	31,625019000000000
20220628_093700_session_1	09:45:05	31,620000000000000	566	51	31,636731	31,625019000000000	25,740000000000000	22,900000000000000	31,625019000000000
20220628_093700_session_1	09:46:05	31,690000000000000	593	59	31,740131000000000	31,705039000000000	25,720000000000000	22,900000000000000	31,705039000000000
20220628_093700_session_1	09:46:05	31,690000000000000	593	59	31,740131000000000	31,705039000000000	25,720000000000000	22,900000000000000	31,705039000000000
20220628_093700_session_1	09:47:05	31,710000000000000	604	62	31,826911000000000	31,745073000000000	25,740000000000000	22,900000000000000	31,745073000000000
20220628_093700_session_1	09:47:05	31,710000000000000	604	62	31,826911000000000	31,745073000000000	25,740000000000000	22,900000000000000	31,745073000000000
20220628_093700_session_1	09:48:05	31,800000000000000	607	67	32,10003	31,890009000000000	25,73	22,900000000000000	31,890009000000000
20220628_093700_session_1	09:48:05	31,800000000000000	607	67	32,10003	31,890009000000000	25,73	22,900000000000000	31,890009000000000
20220628_093700_session_1	09:48:58	31,880000000000000	580	69	32,279412000000000	31,999824000000000	25,710000000000000	23,700000000000000	31,999824000000000
20220628_093700_session_1	09:48:58	31,880000000000000	580	69	32,279412000000000	31,999824000000000	25,710000000000000	23,700000000000000	31,999824000000000
20220628_093700_session_1	09:49:58	31,970000000000000	574	77	32,335649000000000	32,079695000000000	25,73	23,700000000000000	32,079695000000000
20220628_093700_session_1	09:49:58	31,970000000000000	574	77	32,335649000000000	32,079695000000000	25,73	23,700000000000000	32,079695000000000
20220628_093700_session_1	09:51:05	32,040000000000000	591	77	32,272527000000000	32,109758000000000	25,690000000000000	23,700000000000000	32,109758000000000
20220628_093700_session_1	09:51:05	32,040000000000000	591	77	32,272527000000000	32,109758000000000	25,690000000000000	23,700000000000000	32,109758000000000

Figure 7: MySQL Climate data table

This already encompasses two out of the four different data types provided by SenseLab and Aachen. The remaining data will be, as previously discussed, stored in the Non-Relational DMS Neo4j.

4.2 Non-Relational Database Management System

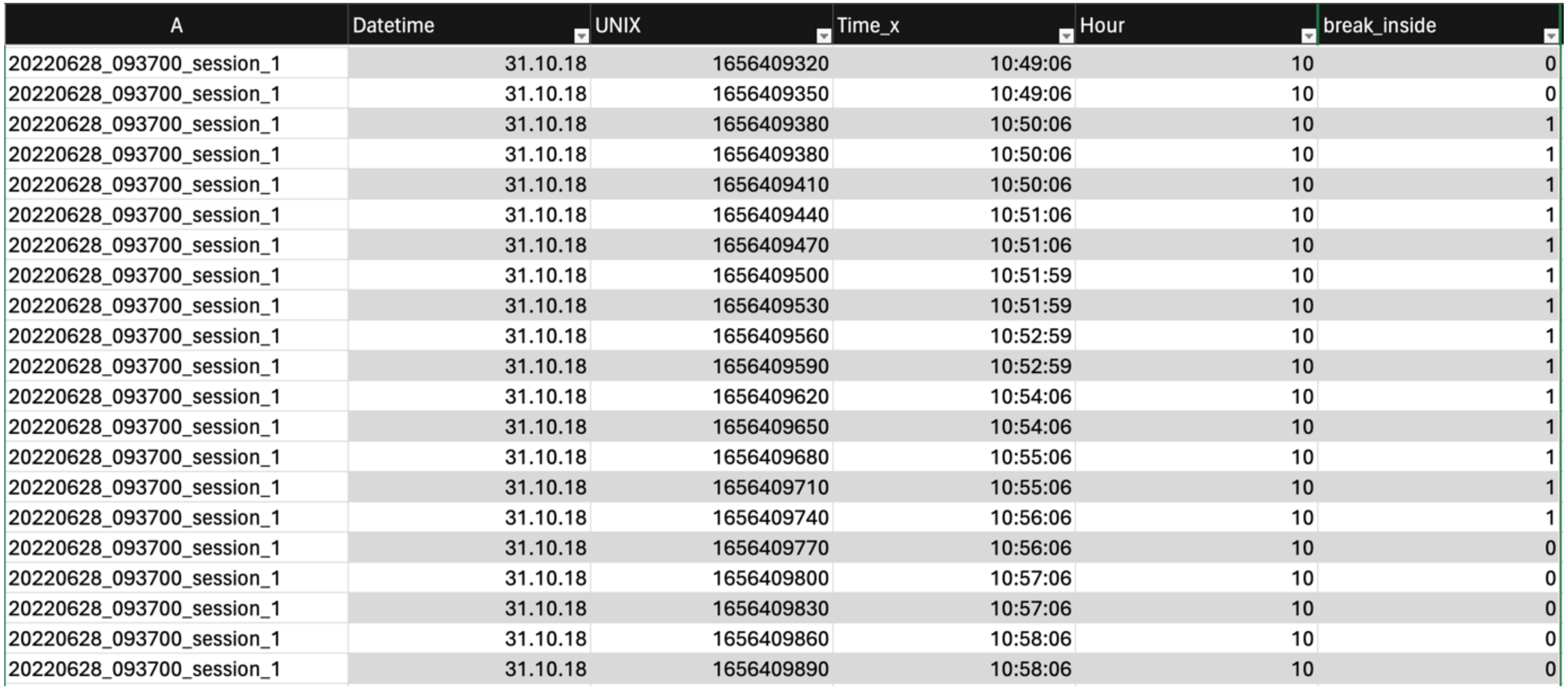
Behavioral data finds its optimal visualization on Neo4j, necessitating preprocessing to align with the intended graph structure. Transforming behavioral data into a graph format offers a distinct advantage, simplifying query processes. Searching for breaks, for example, in a graphical representation, is notably more straightforward than utilizing SQL queries, especially when analyzing the sequence of “ones” as output.

4.2.1 Preprocessing

Before processing, it is crucial to isolate the behavioral data from the provided “.csv” file. It is stored alongside Climate and Physiological Data as a Binary time series. For instance, breaks taken by participants are recorded as "1", assigned to the specific timestamp where the break was taken, and "0" for time slots when breaks were not taken. It is crucial to differentiate between the different types of behavioral data, such as Breaks Inside, Outside, and Other, and the different food and drink intake.

The original CSV file, reduced to relevant columns for this DMS section, is depicted in the following figure. For clarity, only the "break_inside" column is represented, but the same format applies to all other Behavioral Data.

“Figure 8” shows a section of the provided “break_inside” data.



A	Datetime	UNIX	Time_x	Hour	break_inside
20220628_093700_session_1	31.10.18	1656409320	10:49:06	10	0
20220628_093700_session_1	31.10.18	1656409350	10:49:06	10	0
20220628_093700_session_1	31.10.18	1656409380	10:50:06	10	1
20220628_093700_session_1	31.10.18	1656409380	10:50:06	10	1
20220628_093700_session_1	31.10.18	1656409410	10:50:06	10	1
20220628_093700_session_1	31.10.18	1656409440	10:51:06	10	1
20220628_093700_session_1	31.10.18	1656409470	10:51:06	10	1
20220628_093700_session_1	31.10.18	1656409500	10:51:59	10	1
20220628_093700_session_1	31.10.18	1656409530	10:51:59	10	1
20220628_093700_session_1	31.10.18	1656409560	10:52:59	10	1
20220628_093700_session_1	31.10.18	1656409590	10:52:59	10	1
20220628_093700_session_1	31.10.18	1656409620	10:54:06	10	1
20220628_093700_session_1	31.10.18	1656409650	10:54:06	10	1
20220628_093700_session_1	31.10.18	1656409680	10:55:06	10	1
20220628_093700_session_1	31.10.18	1656409710	10:55:06	10	1
20220628_093700_session_1	31.10.18	1656409740	10:56:06	10	1
20220628_093700_session_1	31.10.18	1656409770	10:56:06	10	0
20220628_093700_session_1	31.10.18	1656409800	10:57:06	10	0
20220628_093700_session_1	31.10.18	1656409830	10:57:06	10	0
20220628_093700_session_1	31.10.18	1656409860	10:58:06	10	0
20220628_093700_session_1	31.10.18	1656409890	10:58:06	10	0

Figure 8: Example of “break inside” data

The participant initiated a break inside at 10:50:06 and ended it at 10:56:06. To represent this information graphically, each break slot is a node with the following attributes:

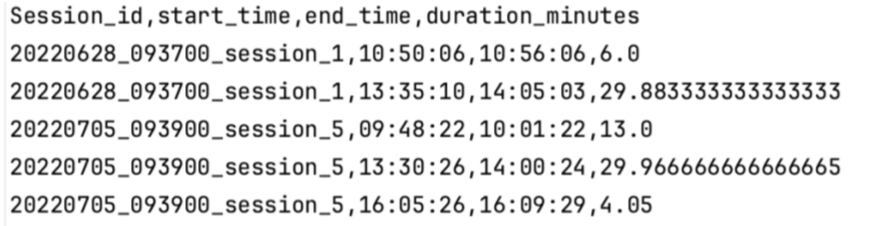
- Start Time
- End Time
- Duration

To facilitate the seamless integration of this data into the Neo4J graph DMS, it is essential to filter out the relevant information from the Time Series CSV data. The most straightforward method to extract the necessary attributes is through a Python script, especially considering that 56 sessions have been conducted so far, each yielding multiple instances of different behavioral data. This script loads the provided “.csv” file data, extracts the sought-after information, and stores it in a new “.csv” file, which serves as the foundation for creating specific behavioral nodes within Neo4J.

The Python script “BehavioralData.py” is responsible for filtering this information out of the complete data set; it iterates through the CSV file, analyzing the different columns to detect changes in sequential time slots. When a transition from "0" to "1" is identified, it marks the start of a break by appending the corresponding timestamp from the "Time_x" column to the specific break list. Similarly, a change from "1" to "0" marks the end of the break. Following this, the code calculates the duration of the specific break and saves this information alongside the other two parameters. It also iterates through the ”.csv” file in order to determine instances where either hot or cold food and drinks were intaken. The code also registers for each of the recorded occurrences the corresponding “session_id”; this is especially helpful when assigning each behavioral event node to the session node in which it happened.

Once the code completes its execution, it stores the recorded breaks and food/drink intakes in a new appropriately named “.csv” file.

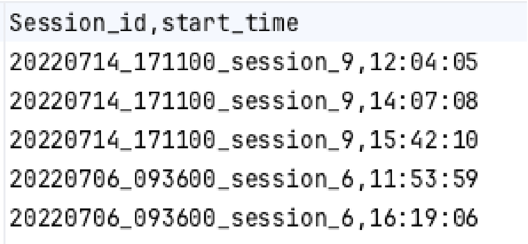
“Figure 9” illustrates how the Breaks Taken inside were stored after being processed by the analyzed code.



```
Session_id,start_time,end_time,duration_minutes
20220628_093700_session_1,10:50:06,10:56:06,6.0
20220628_093700_session_1,13:35:10,14:05:03,29.883333333333333
20220705_093900_session_5,09:48:22,10:01:22,13.0
20220705_093900_session_5,13:30:26,14:00:24,29.966666666666665
20220705_093900_session_5,16:05:26,16:09:29,4.05
```

Figure 9: Break Inside Data frame after preprocessing

“Figure 10” illustrates how the Cold drink events are stored after being processed by the previously introduced code.



```
Session_id,start_time
20220714_171100_session_9,12:04:05
20220714_171100_session_9,14:07:08
20220714_171100_session_9,15:42:10
20220706_093600_session_6,11:53:59
20220706_093600_session_6,16:19:06
```

Figure 10: Cold drink data frame after preprocessing

Notice that in the case of Cold Drink events, unlike breaks, the duration and end times are not included in the output. This is because Cold Drink events are momentary events with only one assigned timestamp. The same principle applies to all food or drink intake events.

4.2.2 Neo4J data-structure

To establish a Non-RDMS within Neo4J, the preprocessed data must be utilized to design the desired structure. The process involves executing Cypher queries to load the preprocessed data into Neo4J, creating nodes, and establishing relationships. The following sections outline the key steps of this process:

The initial step involves loading the Session data extracted from the “.csv” file using the Python script “Sessions.py” into Neo4J, creating a session knot for each of the recorded sessions with the corresponding “session_id” as an attribute.

```
LOAD CSV WITH HEADERS FROM 'file:///sessions.csv' AS row
CREATE (:Session {id: row.Session_id})
```

The next step involves loading breaks and food/ drink intake events data into Neo4J. This is achieved through Cypher queries similar to the one provided for “breaks_inside”. This query ensures that for each break and event type, a knot is created with the correct attributes (“session_id”, “start_time”, “end_time”, “duration_in_minutes”). For each of the created knots, a relationship is simultaneously established to the specific session in which the event happened.

```
// Load breaks inside data
LOAD CSV WITH HEADERS FROM 'file:///breaks_inside.csv' AS row
MATCH (session:Session {id: row.Session_id})
CREATE (:Break:BreakInside {
session_id: row.Session_id,
start_time: row.start_time,
end_time: row.end_time,
duration_minutes: toInteger(row.duration_minutes)
})-[:PARTICIPANT_TAKES_BREAK_INSIDE]->(session);
```

To determine if the created nodes and relationships were correctly implemented, it is important to visualize and analyze the created graph. This is achieved by querying the code displayed next. In this case, the graph for the specific session "20220714_171100_session_9" is called upon.

```
// Visualize the session and its relationships
MATCH path = (session:Session {id: '20220714_171100_session_9'})-[*]-(relatedNode)
RETURN path;
```

Finally, the "session_id" attribute can be removed from the individual behavioral events, as it is already stored within the session node.

"Figure 11" illustrates all sessions and their respective behavioral events.

This structured approach ensures the creation of a meaningful and interconnected graph within Neo4J, representing the behavioral data captured during the Senselab experiment.

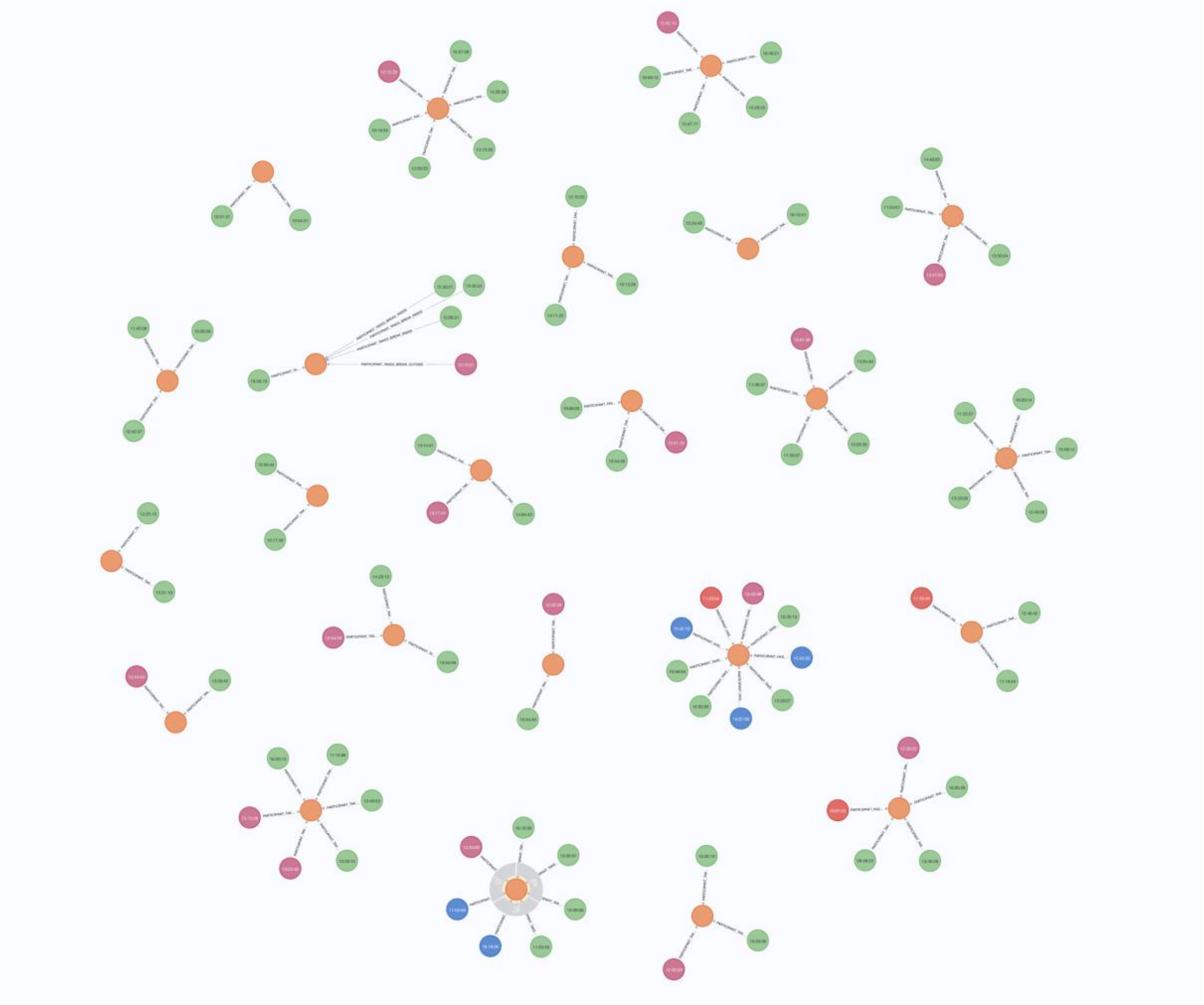


Figure 11: Neo4J DMS all sessions overview

In “Figure 12”, a single session is depicted. Breaks taken inside the experiment chamber are green, while those taken outside are marked in pink. Additionally, hot and cold drinks are represented in red and blue, respectively. Notably, during this session, no food was consumed, neither hot nor cold.

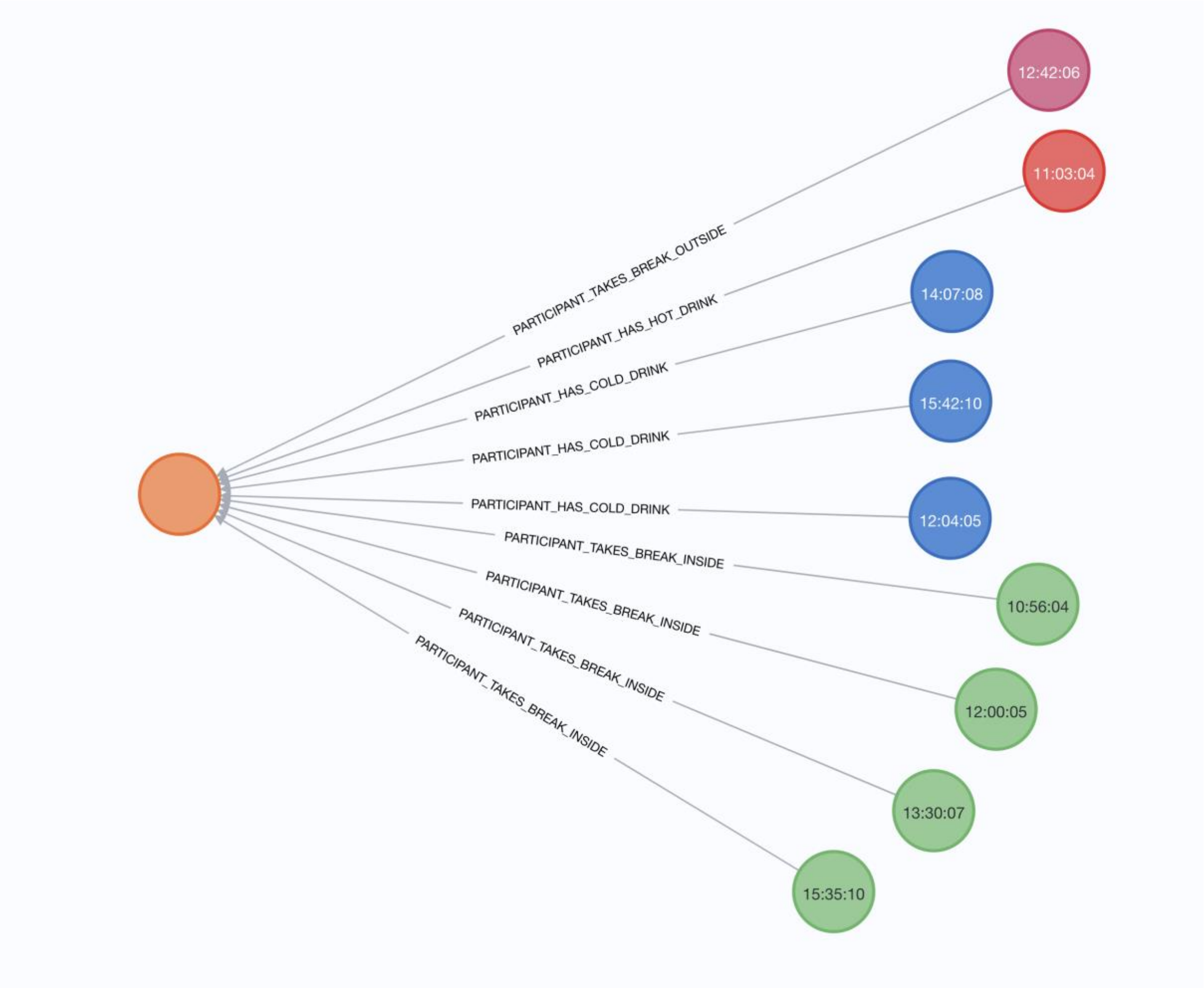

12:42:06
11:03:04
14:07:08
15:42:10
12:04:05
10:56:04
12:00:05
13:30:07
15:35:10
PARTICIPANT_TAKES_BREAK_OUTSIDE
PARTICIPANT_HAS_HOT_DRINK
PARTICIPANT_HAS_COLD_DRINK
PARTICIPANT_HAS_COLD_DRINK
PARTICIPANT_HAS_COLD_DRINK
PARTICIPANT_TAKES_BREAK_INSIDE
PARTICIPANT_TAKES_BREAK_INSIDE
PARTICIPANT_TAKES_BREAK_INSIDE
PARTICIPANT_TAKES_BREAK_INSIDE

Figure 12: Single session graph representation

4.3 Integration of Data Management Systems

Combining the two newly filled DMSs presents an opportunity to best leverage their individual strengths. Integrating them allows for more comprehensive analyses, such as cross-domain analysis, to be performed.

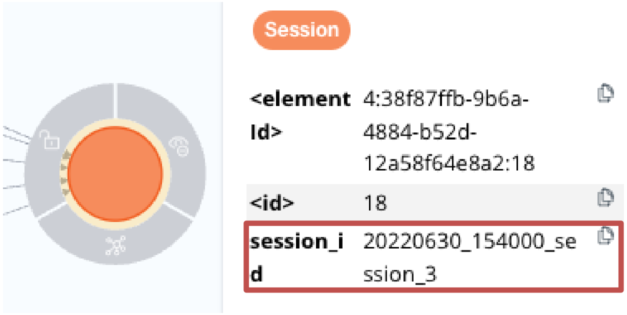
There are several approaches to achieve this integration. One method involves writing a Python script that combines the two DMSs based on specific connection points, such as retrieving climate data based on behavioral events. A detailed example of such integration is explored in the following chapters 5.1.1 and 5.2.1.

Another approach is to implement a variation of the first method using the Dash library for Python. This approach offers the flexibility for users to run analyses based on specific inputs by developing an application capable of plotting graphs based on these chosen inputs.

To develop such an application, it's crucial to first determine the necessary inputs from the user to generate the desired output.

For better data analysis, two different interfaces were developed: one for cross-domain analysis between climate and behavioral data and another for physiological and behavioral data.

These interfaces are stored within the "appClimateData.py" and "appPhysiologicalData.py" Python scripts, respectively. Upon running the script and opening the web app, the user is prompted to provide inputs. Firstly, selecting a session by typing in its specific session ID, easily retrievable from the Neo4J interface, as illustrated in “Figure 13”.



Session

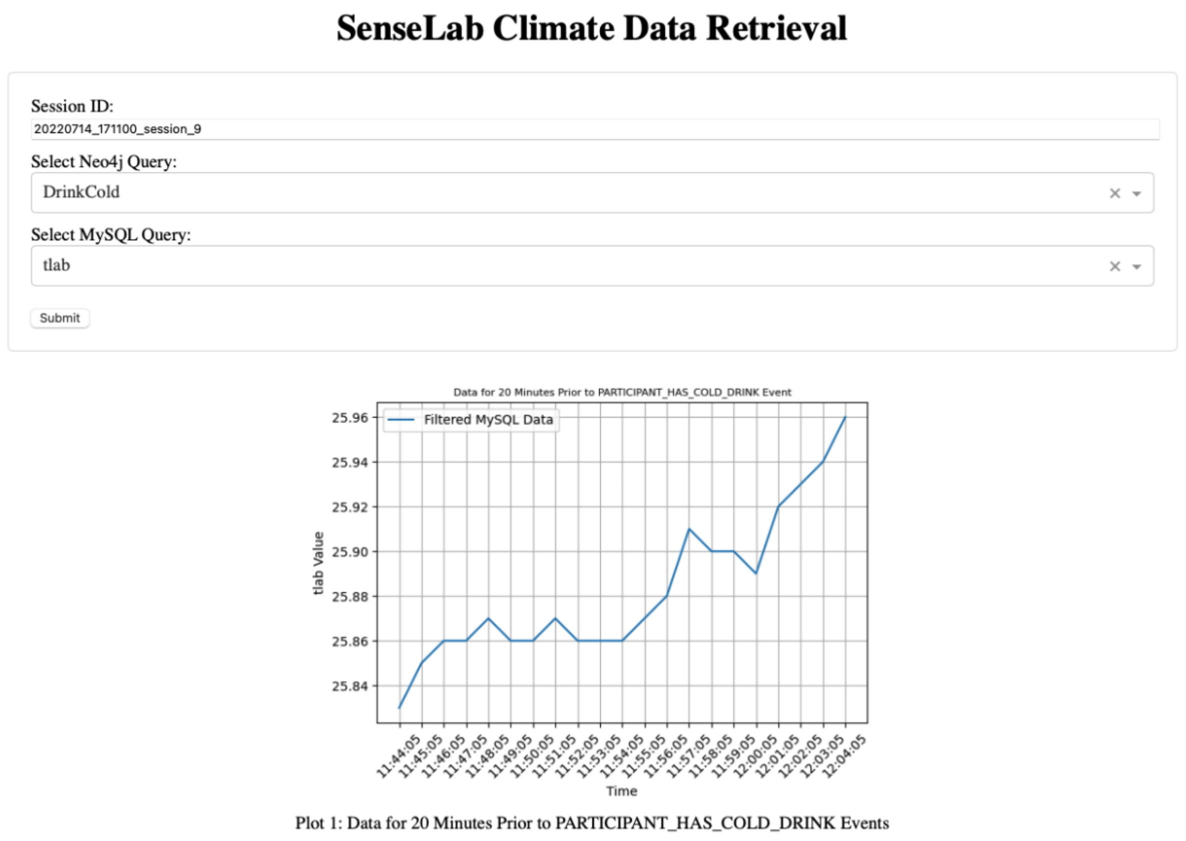
<element Id>	4:38f87ffb-9b6a-4884-b52d-12a58f64e8a2:18
<id>	18
session_id	20220630_154000_session_3

Figure 13: "session_id" attribute

Following this step, the user selects from two drop-down menus: the first to choose the behavioral event for analysis, and the second to select the data to be collected from MySQL. After inputting all necessary information and clicking on the submit button, the script retrieves the start time from the specified event in Neo4J, subtracts 20 minutes, and retrieves the corresponding data from MySQL. It then plots the collected information on the web page interface.

If a behavioral event occurs multiple times during the session, the script iterates through the different start times and plots the corresponding graphs accordingly. For example, the user can retrieve the temperature inside the lab during the 20 minutes prior to the participant drinking something cold.

The interface of the web page, after being filled out and having a graph plotted, is illustrated in “Figure 14”.



SenseLab Climate Data Retrieval

Session ID: 20220714_171100_session_9

Select Neo4j Query: DrinkCold

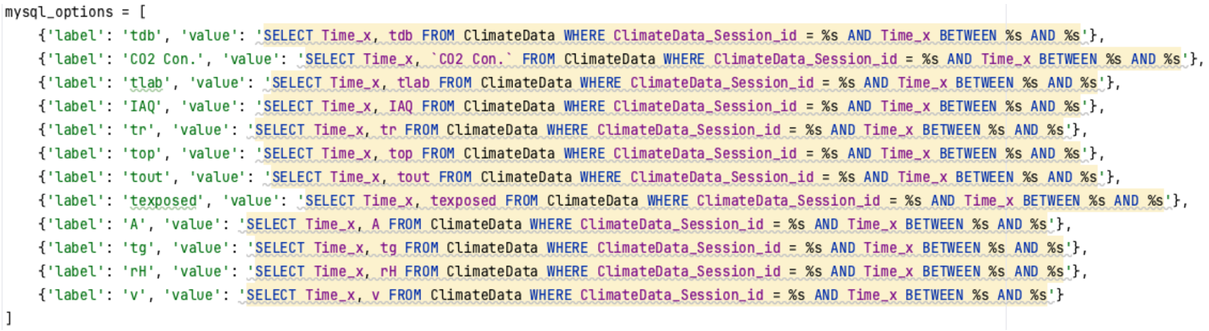
Select MySQL Query: tlab

Plot 1: Data for 20 Minutes Prior to PARTICIPANT_HAS_COLD_DRINK Events

Figure 14: "appClimateData.py" interface

The similar application “appPhysiologicalData.py” functions in a similar manner but plots physiological data instead of climate data.

Using drop-down menus instead of requiring users to type queries for Neo4J and MySQL is advantageous as it eliminates the need for users to be familiar with Cypher for Neo4J and SQL for MySQL. This was accomplished by predefining every possible query in the code and assigning these options to the respective drop-down menus, as shown in “Figure 15”.



```
mysql_options = [
    {'label': 'tdb', 'value': 'SELECT Time_x, tdb FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'CO2 Con.', 'value': 'SELECT Time_x, `CO2 Con.` FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'tlab', 'value': 'SELECT Time_x, tlab FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'IAQ', 'value': 'SELECT Time_x, IAQ FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'tr', 'value': 'SELECT Time_x, tr FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'top', 'value': 'SELECT Time_x, top FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'tout', 'value': 'SELECT Time_x, tout FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'texposed', 'value': 'SELECT Time_x, texposed FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'A', 'value': 'SELECT Time_x, A FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'tg', 'value': 'SELECT Time_x, tg FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'rH', 'value': 'SELECT Time_x, rH FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'},
    {'label': 'v', 'value': 'SELECT Time_x, v FROM ClimateData WHERE ClimateData_Session_id = %s AND Time_x BETWEEN %s AND %s'}
]
```

Figure 15: MySQL Climate data drop-down menu options

5 Case Study

This chapter explores the practical implications of implementing the hybrid DMS in SenseLab's experimental setting. The system's functionality is evaluated through targeted query scenarios, assessing its effectiveness in handling diverse datasets.

The selected query scenarios encompass cross-domain analyses and the detection of behavioral observations. The goal is to uncover both the strengths and limitations of the hybrid DMS in navigating SenseLab's diverse data spectrum.

Results from these scenarios are presented, highlighting key findings and insights. Grounded in the initial research questions, the discussion offers a coherent evaluation of whether the hybrid approach aligns with SenseLab's practical needs.

5.1 Selected Query Scenarios

In this section, the selected query scenarios are examined to assess the functionality and effectiveness of the hybrid DMS within the context of SenseLab's data requirements. Each scenario addresses possible challenges identified previously in this thesis.

5.1.1 Scenario 1: Cross-Domain Analysis

Challenge: Evaluate the hybrid DMS's ability to perform cross-domain analysis by correlating climate conditions, participant behavior, and physiological responses.

Example: If the participant drinks water, what was the climate condition during the 20 minutes prior to that?

Query: Retrieve and analyze data to identify correlations between changes in climate conditions, participant behavior patterns, and corresponding physiological responses over the experiment's duration.

5.1.2 Scenario 2: Behavioral Data Exploration

Challenge: Visualizing the behavioral data within the DMS.

Example: Analyze the different breaks taken by a participant during a specific session.

Query: Create an intuitive visual representation to showcase the breaks taken by participants during a particular session. Utilize visual search techniques instead of relying on traditional SQL queries to enhance the user experience and make data exploration more accessible.

5.2 Implementing

In this section, a thorough analysis of the selected scenarios is conducted, highlighting the benefits derived from utilizing the developed hybrid DMS. Through these analyses, the aim is to explain how integrating different data sources in SenseLab's experiments brings practical benefits and efficiencies.

5.2.1 Scenario 1: Cross-Domain Analysis

To effectively harness the benefits of cross-domain analysis through the hybrid DMS, it is important to combine them in a manner that enables seamless data retrieval and analysis across platforms. In the context of SenseLab's data, this mostly means retrieving behavioral data and utilizing it to analyze either climate or physiological data. The integration of the two DMSs has been previously discussed in Chapter 4.3.

This section of the thesis explores the advantages of the presented integration method. The first approach involves developing a Python script capable of retrieving data from Neo4J DMS and using it to perform analysis on MySQL data. For instance, consider a script that searches for "Hot Drink" events within a specific session in Neo4j. Once the data is located, the script extracts the Start Time attribute from the event node and retrieves temperature data from MySQL for the 20 minutes preceding this event. The resulting data is then visualized using Matplotlib.

The script, named "tlab.py", offers precise analysis of the searched information. However, it lacks flexibility in terms of changing parameters without modifying specific parts of the script or using multiple scripts to retrieve different data.

The plotted data from the "tlab.py" script is depicted in the following figure.

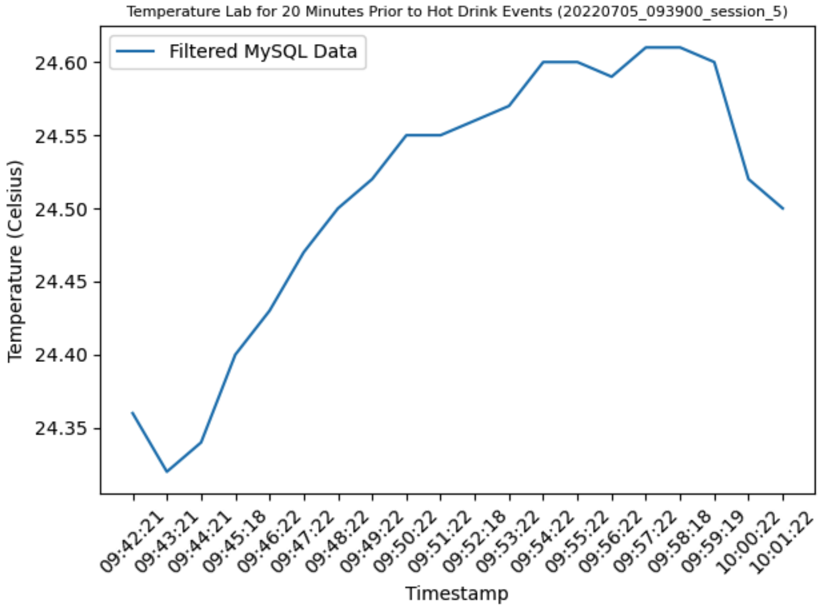


Figure 16: tlab plot before the Hot Drink event

Alternatively, a more efficient approach involves utilizing previously presented apps such as "appClimateData.py" and "appPhysiologicalData.py." These apps allow for the analysis of multiple climate and physiological parameters across various behavioral events and sessions without the need to exit the interface. This approach leverages the same benefits presented by using “tlab.py” while enhancing flexibility and streamlining the analysis process.

5.2.2 Scenario 2: Behavioral Data Exploration

If behavioral data were integrated within the MySQL data scope, retrieval of this information would involve a simple SQL query, such as:

```
SELECT *

FROM BehavioralData

WHERE session_id = '20220714_171100_session_9' AND breakinside = 1;
```

Figure 17: Possible SQL Query

This query would provide a sequential list of “1s”, marking the minutes where breaks were taken inside the chamber within the specific session, along with their corresponding timestamps. While effective for analytical purposes, implementing behavioral data within a GraphDMS like Neo4J offers a more visually appealing and user-friendly representation.

Similar to the SQL query, specific sessions can be called upon using the cypher language in Neo4J based on their unique “session_id”, streamlining the search process. Once the Session node is retrieved, users can visually locate the desired break inside node. By opening this node, users can access its attributes to establish the start and end times of the break and its duration.

Neo4J's capabilities extend further by allowing users to color-code nodes and adjust their sizes, enhancing the intuitive nature of the exploration process. For example, users can query a desired session using the following code:

```
MATCH path = (session:Session {session_id: '20220714_171100_session_9'})-[*]-
(relatedNode)
RETURN path;
```

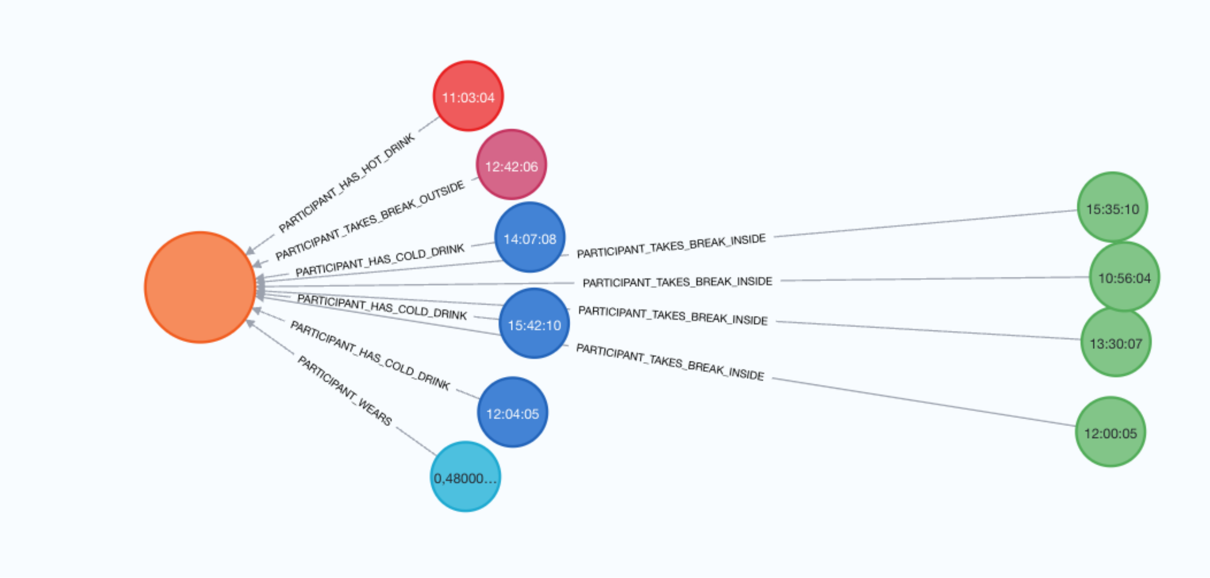
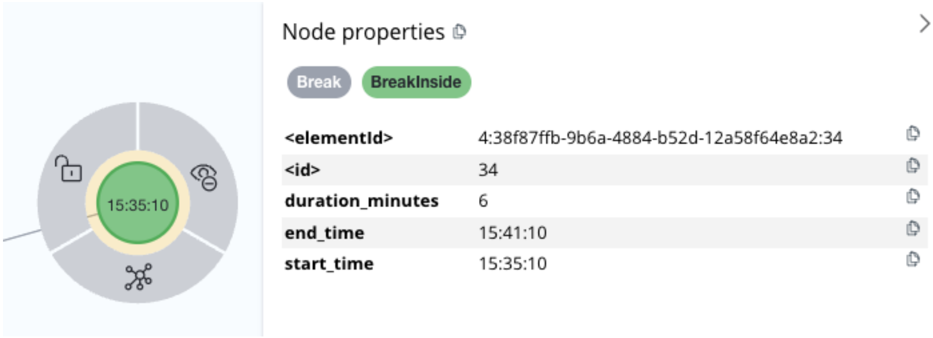

11:03:04
12:42:06
14:07:08
15:42:10
12:04:05
0,48000...
15:35:10
10:56:04
13:30:07
12:00:05
PARTICIPANT_HAS_HOT_DRINK
PARTICIPANT_TAKES_BREAK_OUTSIDE
PARTICIPANT_HAS_COLD_DRINK
PARTICIPANT_HAS_COLD_DRINK
PARTICIPANT_HAS_COLD_DRINK
PARTICIPANT_WEARS
PARTICIPANT_TAKES_BREAK_INSIDE
PARTICIPANT_TAKES_BREAK_INSIDE
PARTICIPANT_TAKES_BREAK_INSIDE
PARTICIPANT_TAKES_BREAK_INSIDE

Figure 18: Session Neo4J and its behavioral events nodes

Neo4J then generates a graph, illustrated in “Figure 18”, with the orange node representing the searched session and other nodes storing different behavioral data. Users can visually search for specific nodes based on color or utilize specifically named relationships to aid in locating the desired node. For instance, green nodes represent breaks taken inside the experiment chamber, with timestamps indicating the start times of each event. By following relationships labeled "PARTICIPANT_TAKES_BREAK_INSIDE," users can reach the Break inside node.

Upon clicking on an event node, users can retrieve necessary information from its attributes, as shown in the following figure:



Node properties

Break BreakInside

<elementId>	4:38f87ffb-9b6a-4884-b52d-12a58f64e8a2:34
<id>	34
duration_minutes	6
end_time	15:41:10
start_time	15:35:10

Figure 19: "BreakInside" node Neo4J

While attributes such as "elementId" and "id" are automatically generated by Neo4J, others were implemented using methods described in Chapter 4.2.

This process ensures that users can quickly retrieve information without the need for multiple queries in MySQL and manual calculation of break durations. By visually exploring the data in Neo4J, users can efficiently identify and access the necessary information, streamlining the analysis process and enhancing overall productivity.

6 Limitations

In this section of the thesis, possible limitations of the developed hybrid DMS are analyzed, highlighting the hurdles that users of this system may encounter while also presenting possible remediations for the limitations encountered.

6.1 Data Range Limitations

The data range of this thesis is limited to the data provided by experiments conducted by SenseLab. While SenseLab collects various types of data across multiple projects, this thesis specifically focuses on the data format and structure adopted by the project at the point in time of this research. Consequently, any data collected by other projects or SenseLab in the future using formats different from those previously adopted may not be seamlessly implemented within the system format presented in this thesis.

SenseLab's wide range of experimental data encompasses multiple recorded parameters, including climate conditions, physiological measurements, and participant behaviors. This thesis addresses the integration and management of these data types within a hybrid DMS; therefore, it does not extend to the incorporation of additional data sources or formats beyond those provided by SenseLab's experiments at this point in time.

The findings and conclusions drawn from this thesis are constrained by the specific scope and context of SenseLab's data and experimental framework.

To address this limitation, researchers must maintain the same data structure adopted by SenseLab for future experiments or address possible encountered problems through modifications of the process presented in this thesis.

6.2 Technical Limitations

The successful implementation and operation of the hybrid DMS within SenseLab's experimental framework are subject to various technical limitations and constraints. This section highlights key challenges associated with software dependencies, compatibility issues, and the management of system resources, along with potential strategies to address these limitations effectively.

6.2.1 Software and Tool Limitations

The hybrid DMS relies on various Python libraries and drivers for data management, analysis and visualization. Technical limitations or constraints associated with these tools may impact the functionality and performance of the system.

1. Software Dependencies and Compatibility:
    - **Description:** The integration of libraries within the Dash applications script, including Matplotlib, Neo4j, and MySQL, introduces possible compatibility concerns. Ensuring seamless operation across different versions requires constant configuration and thorough testing.
    - **Implications:** Incompatibility between different components can lead to errors, inconsistencies, and decreased performance. Managing the different libraries and drivers effectively is crucial to maintaining system stability and reliability.
    - **Possible solution:** Regular updates and version control can help address compatibility issues. Constant testing of the system enables early detection of compatibility issues and facilitates a fast resolution.
2. Localhosting and Multi-user Access:
    - **Description:** Hosting the DMS on a local server restricts access to a single user or device at a time, limiting collaboration opportunities. Opening the system to simultaneous multi user access poses technical challenges, most notably data synchronization.
    - **Implications:** Simultaneous access by multiple users may lead to contention for system resources, performance decrease, and potential data integrity issues. Lack of synchronization mechanisms can result in conflicts and inconsistencies in data processing and analysis.
    - **Possible solution:** Implementing a centralized server architecture allows multiple users to access the system and opens the door for efficiently addressing the encountered issues.

6.2.2 Future Considerations

Addressing technical limitations requires ongoing monitoring and adaptation to enhance the functionality and performance of the hybrid DMS. Constant evaluation of software tools is essential for solving technical constraints and ensuring the long-term effectiveness of the system.

By identifying and addressing technical limitations, SenseLab can optimize the hybrid DMS to meet evolving data management needs and support seamless integration with the existing experimental framework. Collaboration with experts and IT specialists facilitates the identification of technical constraints and the implementation of effective solutions.

7 Summary

In this thesis, the development and implementation of a hybrid DMS tailored to the data requirements of SenseLab's experiments was explored. Through a comprehensive analysis of relational and non RDMS architectures, as well as practical case studies and query scenarios, the benefits, challenges, and limitations of integrating different data sources within a single framework were made evident.

7.1 Key Findings

- Architecture Selection: The decision to blend relational and GraphDMS was driven by the heterogeneous nature of SenseLab's data, allowing the use of the strengths of each model to manage diverse datasets effectively.

- Requirements Analysis: By categorizing present data use and future considerations, essential criteria for selecting and evaluating the DMS were identified, including data integration, management, performance and future-proofness.

- Development of DMS: The structure of both relational and non-relational components was meticulously designed, utilizing Entity Relationship Diagrams and Cypher queries to represent the data structure in the best possible way.

- Case Studies and Query Scenarios: Through targeted query scenarios, the functionality of the hybrid DMS in handling cross-domain analyses and behavioral data exploration was evaluated. Practical implementations demonstrated the system's effectiveness in retrieving, analyzing, and visualizing diverse datasets.

7.2 Limitations and Future Directions

- Data Range Limitations: While the thesis provides insights into SenseLab's experimental data, its scope is limited to the specific data format and structure adopted by SenseLab. Future research could explore the integration of eventual additional data formats.

- Technical Limitations: Challenges related to software dependencies, compatibility issues, and multi-user access pose difficulties to the system's integrity and usability. Addressing these limitations requires continuous evaluation and adaptation.

7.3 Conclusion

In conclusion, the hybrid DMS presented in this thesis offers a robust framework for managing, analyzing, and visualizing heterogeneous datasets in SenseLab's present experimental setting. By integrating relational and non-relational DB, the system's capability to handle diverse data types and facilitate cross-domain analyses was demonstrated. Despite limitations, proactive management and adaptation are key for optimizing the system's functionality and ensuring its long-term effectiveness in supporting SenseLab’s experimental research endeavor.

Bibliography

[1] B. Kobas and S. Clark Koth, ‘SenseLab Webpage’, SenseLab Quantifying Spatial Well-being. Accessed: Nov. 01, 2023. [Online]. Available: https://www.arc.ed.tum.de/klima/forschung/forschungslabore/senselab/

[2] Copyright ASHRAE, ‘ASHRAE Webpage’, ASHRAE’s Mission and Vision. Accessed: Jan. 12, 2024. [Online]. Available: https://www.ashrae.org/about/mission-and-vision

[3] B. Kobas, S. C. Koth, K. Nkurikiyeyezu, G. Giannakakis, and T. Auer, ‘Effect of Exposure Time on Thermal Behaviour: A Psychophysiological Approach’, Chair of Building Technology and Climate Responsive Design, Department of Architecture, Technical University of Munich, 80333 Munich, Germany, Technical University of Munich, 80333 Munich, Germany, 2021. doi: 10.3390/signals2040050.

[4] B. Kobas, S. C. Koth, and T. Auer, ‘TOWARDS A MULTIVARIATE TIME-SERIES APPROACH WITH BIOSIGNAL DATASETS FOR THE GLOBAL THERMAL COMFORT DATABASE’, München, 2022.

[5] V. Földváry Ličina *et al.*, ‘Development of the ASHRAE Global Thermal Comfort Database II’, *Build Environ*, vol. 142, pp. 502–512, Sep. 2018, doi: 10.1016/j.buildenv.2018.06.022.

[6] M. Nasar and M. A. Kausar, ‘Suitability Of Influxdb Database For IoT Applications’, *International Journal of Innovative Technology and Exploring Engineering*, Aug. 2019.

[7] T. Oetike, ‘RRDtool Webpage’, rrdtool. Accessed: Feb. 12, 2024. [Online]. Available: https://oss.oetiker.ch/rrdtool/doc/rrdtool.en.html

[8] B. Kobas, ‘Physiological reactions under different climatic conditions with similar PMV values throughout the workday’, Munich.

[9] B. Kobas, S. Clark Koth, and Prof. Thomas Auer, ‘TUMSENSELAB Quantifying Spatial Wellbeing’.

[10] C. Györödi, R. Gyrörödi, G. Perchele, and A. Olah, ‘A Comparative Study: MongoDB vs. MySQL’, 13th International Conference on Engineering of Modern Electric Systems (EMES), 2015.

[11] C. Nance, T. Losser, R. Iype, and G. Harmon, ‘NOSQL VS RDBMS - WHY THERE IS ROOM FOR BOTH’, 2013. [Online]. Available: http://aisel.aisnet.org/sais2013

[12] D. Yertay, ‘Overview of the method for choosing the most suitable database system according to certain criteria’, in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Sep. 2020. doi: 10.1145/3410352.3410788.

[13] D. Fernandes and J. Bernardino, ‘Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB’, Copyright © 2018 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved, 2018. doi: 10.5220/0006910203730380.

[14] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, *A Comparison of a Graph Database and a Relational Database*. Association for Computing Machinery, 2010.

[15] S. K. Jensen, T. B. Pedersen, and C. Thomsen, ‘Time Series Management Systems: A Survey’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11. IEEE Computer Society, pp. 2581–2600, Nov. 01, 2017. doi: 10.1109/TKDE.2017.2740932.

[16] ‘SQL Join - sqlguides Webpage’. Accessed: Dec. 16, 2023. [Online]. Available: https://www.sqlguides.com/sql_join.php

[17] C. Strauch and W. Kriha, ‘NoSQL Databases’, Stuttgart.

[18] D. Singh Rawat and N. Navneet Kashyap, ‘Graph Database: A Complete GDBMS Survey’, *IJIRST-International Journal for Innovative Research in Science & Technology*|, vol. 3, May 2017, [Online]. Available: www.ijirst.org

[19] M. Knight, ‘dataversity Webpage’, What Is BASE? Accessed: Feb. 12, 2024. [Online]. Available: https://www.dataversity.net/what-is-base/

[20] ‘MySQL Workbench - mysql Webpage’. Accessed: Jan. 03, 2024. [Online]. Available: https://dev.mysql.com/doc/workbench/en/wb-intro.html

[21] J. Pokorny, ‘Graph Databases: Their Power and Limitations’, Prague, Czech Republic. [Online]. Available: http://www.springer.com/series/7409

[22] I. Robinson, J. Webber, and E. Eifrem, ‘Graph Databases’, May 2015.

[23] J. Barrasa, ‘Youtube’, Going Meta - Ep 13: Creating (and RDF-izing) virtual graphs over external data. Accessed: Dec. 12, 2023. [Online]. Available: https://www.youtube.com/watch?v=FoHAyBhcH4s

[24] F. Pfitzner, A. Braun, and A. Borrmann, ‘Object Detection Based Knowledge Graph Creation: Enabling Insight into Construction Processes’.

[25] J. Wu and J. Schlenger, ‘Enriching Building Graphs with Parametric Design Constraints for Automated Design Adaptation’, *Forum Bauinformatik*, 2023, doi: 10.13154/294-10116.

[26] S. Sumathi and S. Esakkirajan, *Fundamentals of Relational Database Management Systems*. Springer, 2007.

[27] ‘“mysql-connector-python” - MySQL Website’. Accessed: Jan. 10, 2024. [Online]. Available: https://dev.mysql.com/doc/connector-python/en/

[28] ‘“neo4j driver” - Neo4J Webpage’. Accessed: Jan. 10, 2024. [Online]. Available: https://neo4j.com/docs/api/python-driver/current/

Appendix

The digital Appendix includes:

- Python scripts used during the preprocessing of the data stage (“BehavioralData.py”, “Sessions.py”, “Start&End.py”, ”clo.py”)
- Python scripts containing the two different developed applications (“appBehavioralData.py", ”appClimateData.py”)
- “.csv” files containing the data provided by SenseLab (“BehavioralDataSenseLab.csv”, ”FullDS.csv”)
- Two dump files containing the Neo4J and MySQL SenseLab Databases (“SenseLab.sql”, “neo4j.dump”)

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelor-Thesis selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ich versichere außerdem, dass die vorliegende Arbeit noch nicht einem anderen Prüfungsverfahren zugrunde gelegen hat.

München, 15. February 2024 [redacted]

[redacted]

[redacted]