Unit 8 Seminar

Requirements

- Functional Requirements
- Student enrolment and deregistration
- Attendance logging
- Grade management
- Timetable generation
- Internal messaging

Non-Functional Requirements

- Fast data access
- Secure student data handling
- Real-time data consistency
- Scalability for future growth
- System reliability

Data Structures

Data Structure	Use Case	Justification
Hash Table	Store and retrieve student records by ID	Constant-time lookups; ideal for key-based data like student IDs
Graph	Model timetable constraints and relationships	Suitable for detecting scheduling conflicts and dependencies

Justification

Python.org describes dictionaries as providing very fast lookups and associative memory. Perfect for unique keys like student IDs.

Graphs handle complex relationships. Timetables involve constraints between rooms, times, staff and students, graphs can represent these effectively.

Abeykoon et al. (2022) promote using well defined, efficient data structures like:

- Tables for tabular data.
- Graphs and distributed operators for workloads that involve relationships.

Their HPTMT architecture shows that selecting optimal data structures improves performance and scalability, especially in systems requiring high concurrency and distributed processing.

Similar Article

The articles by Abeykoon et al. (2022) and Perera et al. (2023) are directly connected. Abeykoon introduces the HPTMT (High Performance Tensors, Matrices and Tables) architecture. It is a scalable model based on parallel operators. The paper also presents Cylon, a framework built to support this model. Perera expands on this by designing distributed memory data frames using the same principles. Both use BSP (Bulk Synchronous Parallel) for consistent performance. They reject asynchronous systems like Spark and Dask. The focus is on compact data structures, tables, arrays and scalars.

References

Abeykoon, V. et al. (2022). *HPTMT Parallel Operators for High Performance Data Science and Data Engineering*. Frontiers in Big Data, 4:756041. https://doi.org/10.3389/fdata.2021.756041

Perera, N. et al. (2023). *High Performance Dataframes from Parallel Processing Patterns*. In: Wyrzykowski, R. et al. (eds) *Parallel Processing and Applied Mathematics*. PPAM 2022. Lecture Notes in Computer Science, vol 13826. Springer. https://doi.org/10.1007/978-3-031-30442-2_22

Python Software Foundation. (n.d.). *The Python language reference (Version 3)*. Retrieved July 18, 2025, from https://docs.python.org/3/reference/index.html