Heterogeneous Database Systems Assignment

Let’s make a heterogeneous database system using Neo4j and Dynamo for academic journal papers.

Dynamo would make a strong choice to store the actual abstracts of the papers because you would get:

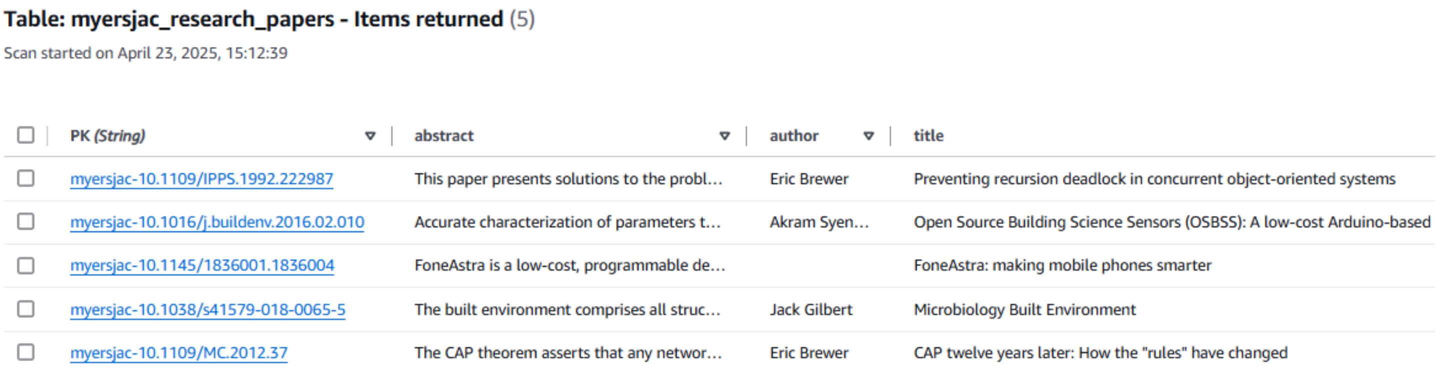


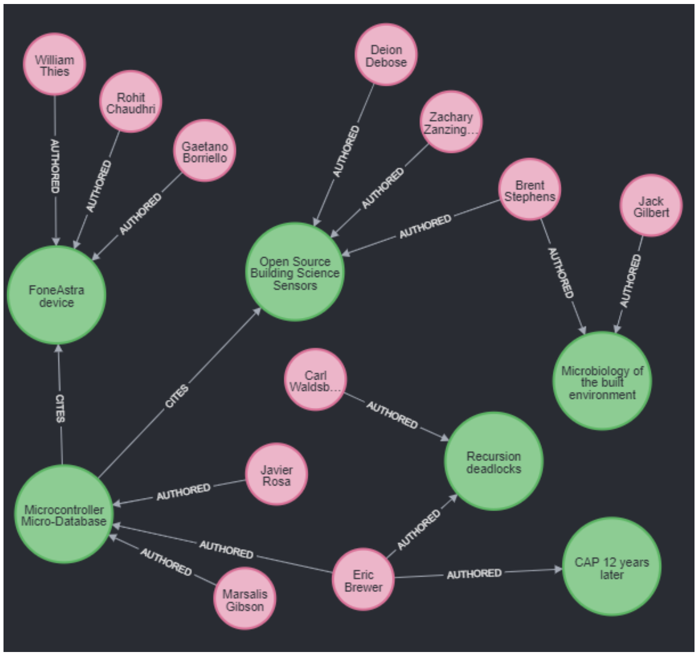
Table: myersjac_research_papers - Items returned (5)

Scan started on April 23, 2025, 15:12:39

☐	PK *(String)*	abstract	author	title
☐	myersjac-10.1109/IPPS.1992.222987	This paper presents solutions to the probl…	Eric Brewer	Preventing recursion deadlock in concurrent object-oriented systems
☐	myersjac-10.1016/j.buildenv.2016.02.010	Accurate characterization of parameters t…	Akram Syen…	Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based
☐	myersjac-10.1145/1836001.1836004	FoneAstra is a low-cost, programmable de…		FoneAstra: making mobile phones smarter
☐	myersjac-10.1038/s41579-018-0065-5	The built environment comprises all struc…	Jack Gilbert	Microbiology Built Environment
☐	myersjac-10.1109/MC.2012.37	The CAP theorem asserts that any networ…	Eric Brewer	CAP twelve years later: How the "rules" have changed

- **High Performance at Scale**: consistent single-digit millisecond latency for read and write operations, regardless of the dataset size. This ensures rapid access to paper metadata, even as your collection grows.
- **Seamless Scalability**: automatic scaling to accommodate varying workloads, eliminating the need for manual provisioning or capacity planning.
- **Flexible Data Modeling**: flexibility to store diverse metadata attributes without predefined schemas, accommodating various fields like authors, publication dates, and keywords.
- **Global Availability**: with support for multi-region replication, low-latency access to data for users worldwide, enhancing the user experience for global research communities.

But Neo4j would also offer some strong advantages, including:



- **Natural Representation of Complex Relationships** as scientific literature is inherently interconnected: papers cite other papers, authors collaborate across institutions, and topics evolve over time. Neo4j’s graph model allows you to represent these relationships directly as nodes and edges, making it intuitive to model and query such data structures.

- **Efficient Traversal of Deeply Nested Data** as traditional relational databases can struggle with queries that require multiple joins, such as finding all papers that cite a given paper, then finding all papers that cite those papers, and so on. Neo4j excels at traversing such deep relationships efficiently, enabling complex queries like:
    - Identifying influential papers within a specific field.
    - Tracing the evolution of research topics over time.
    - Discovering collaboration networks among researchers.

- **Advanced Graph Algorithms for Insight Extraction** that can be applied to analyze the research network to determine the most influential papers based on citation networks, to identify clusters of related research topics or author groups, or to find papers or authors with similar profiles or research.

- **Visualization and Explorations** to enable interactive exploration of the research graph.

Structure of databases

Neo4j – all nodes should look like this:

```
{
  "identity": 120,
  "labels": [
    "Paper"
  ],
  "properties": {
"name": "SNOW Revisited",
"title": "SNOW Revisited: Understanding When Ideal READ Transactions Are Possible",
"createdBy": "myersjac",
"year": "2021",
"doi": "myersjac-10.1109/IPDPS49936.2021.00101"
  }
}
```

```
{
  "identity": 119,
  "labels": [
    "Author"
  ],
  "properties": {
"name": "Nancy Lynch",
"createdBy": "myersjac"
  }
}
```

Relationships

Primary Author

```
{
  "identity": 19,
  "start": 159,
  "end": 120,
  "type": "AUTHORED",
  "properties": {
"primary": true
  }
}
```

Secondary Authors

```
{
  "identity": 22,
  "start": 119,
  "end": 120,
  "type": "AUTHORED",
  "properties": {

  }
}
```

Citations

```
{
  "identity": 16,
  "start": 152,
  "end": 141,
  "type": "CITES",
  "properties": {

  }
}
```

Dynamo

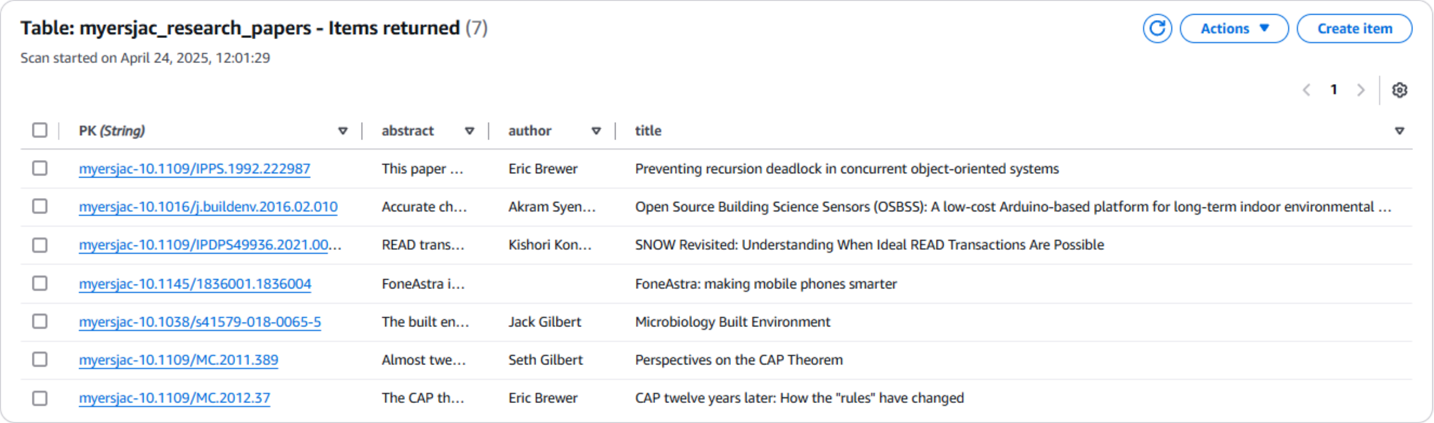


Table: myersjac_research_papers - Items returned (7)

Scan started on April 24, 2025, 12:01:29

□	PK *(String)*	abstract	author	title
□	myersjac-10.1109/IPPS.1992.222987	This paper …	Eric Brewer	Preventing recursion deadlock in concurrent object-oriented systems
□	myersjac-10.1016/j.buildenv.2016.02.010	Accurate ch…	Akram Syen…	Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental …
□	myersjac-10.1109/IPDPS49936.2021.00…	READ trans…	Kishori Kon…	SNOW Revisited: Understanding When Ideal READ Transactions Are Possible
□	myersjac-10.1145/1836001.1836004	FoneAstra i…		FoneAstra: making mobile phones smarter
□	myersjac-10.1038/s41579-018-0065-5	The built en…	Jack Gilbert	Microbiology Built Environment
□	myersjac-10.1109/MC.2011.389	Almost twe…	Seth Gilbert	Perspectives on the CAP Theorem
□	myersjac-10.1109/MC.2012.37	The CAP th…	Eric Brewer	CAP twelve years later: How the "rules" have changed

Dynamo stores the primary key as the doi number (preceded by your username), the abstract of a paper, and the primary author only.

Neo4J stores all authors, an abbreviated title for the name (as well as the full title), and no abstracts.

Look at the sample code and finish any unfinished coding!