**Batch T4**

**Practical No. 11**

**Title of Assignment : Neo4j Graph Database**

**Student Name: Parshwa Herwade**

**Student PRN: 22510064**

**Objective / Aim**

* To model a research‑papers domain as a property graph in Neo4j, capturing authorship, hierarchical classifications, and citation relationships.
* To import the Cora dataset (~37K papers, 25K authors, 220K relationships) into Neo4j via the Data Browser.
* To design an Angular-based search form enabling:
  1. Multi‑level citation checks (does paper A cite paper B directly or transitively?).
  2. Retrieval of the full classification hierarchy for any paper.
  3. Execution of additional custom queries.

**Introduction**  
Knowledge of how research papers connect—via shared authors, topic hierarchies, and citations—is critical for literature surveys and impact analysis. Neo4j’s labeled property‑graph model naturally represents these interlinked entities, enabling efficient traversal queries and path‑finding. An Angular front end can expose this graph’s capabilities through dynamic search forms.

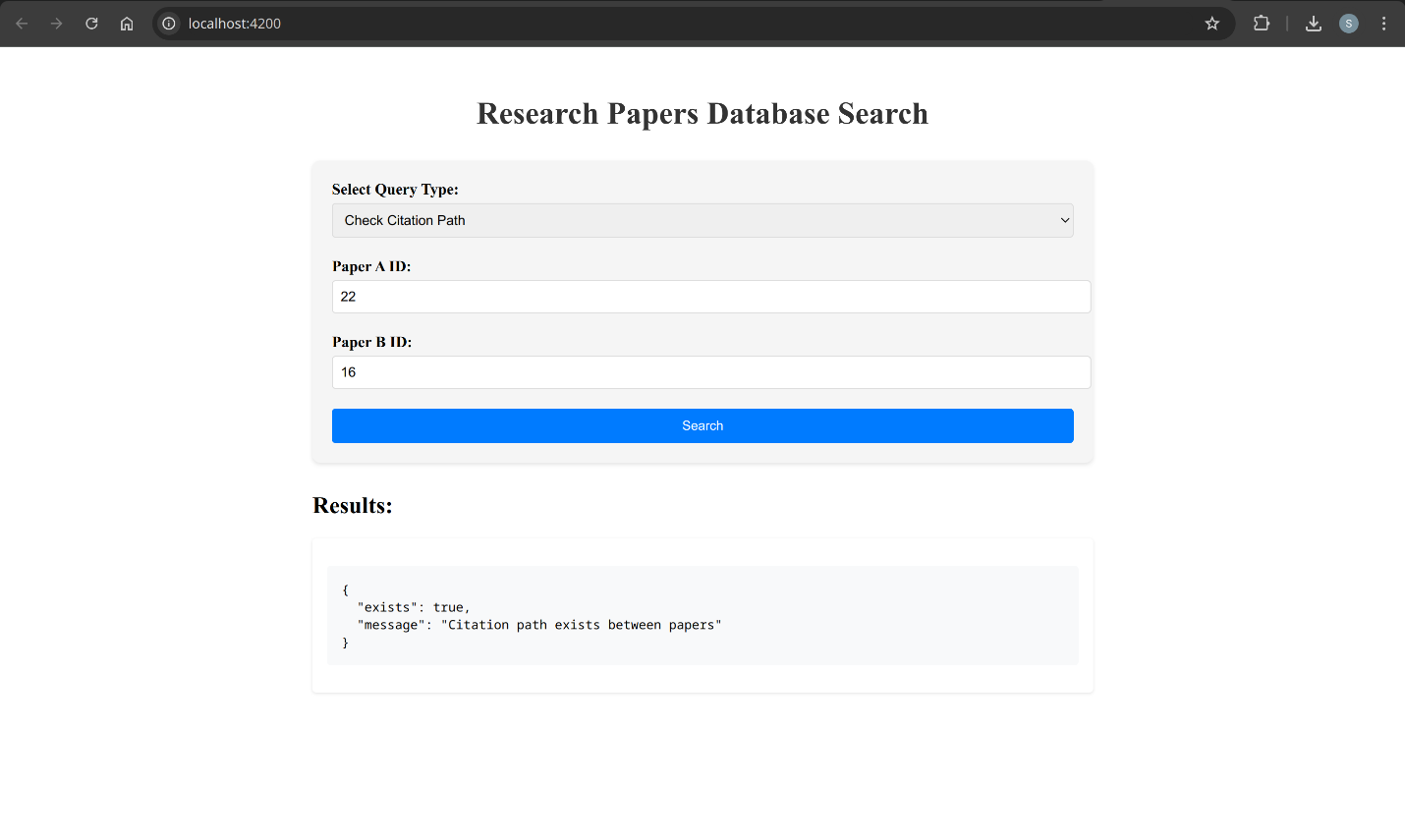
**Theory / Algorithms**

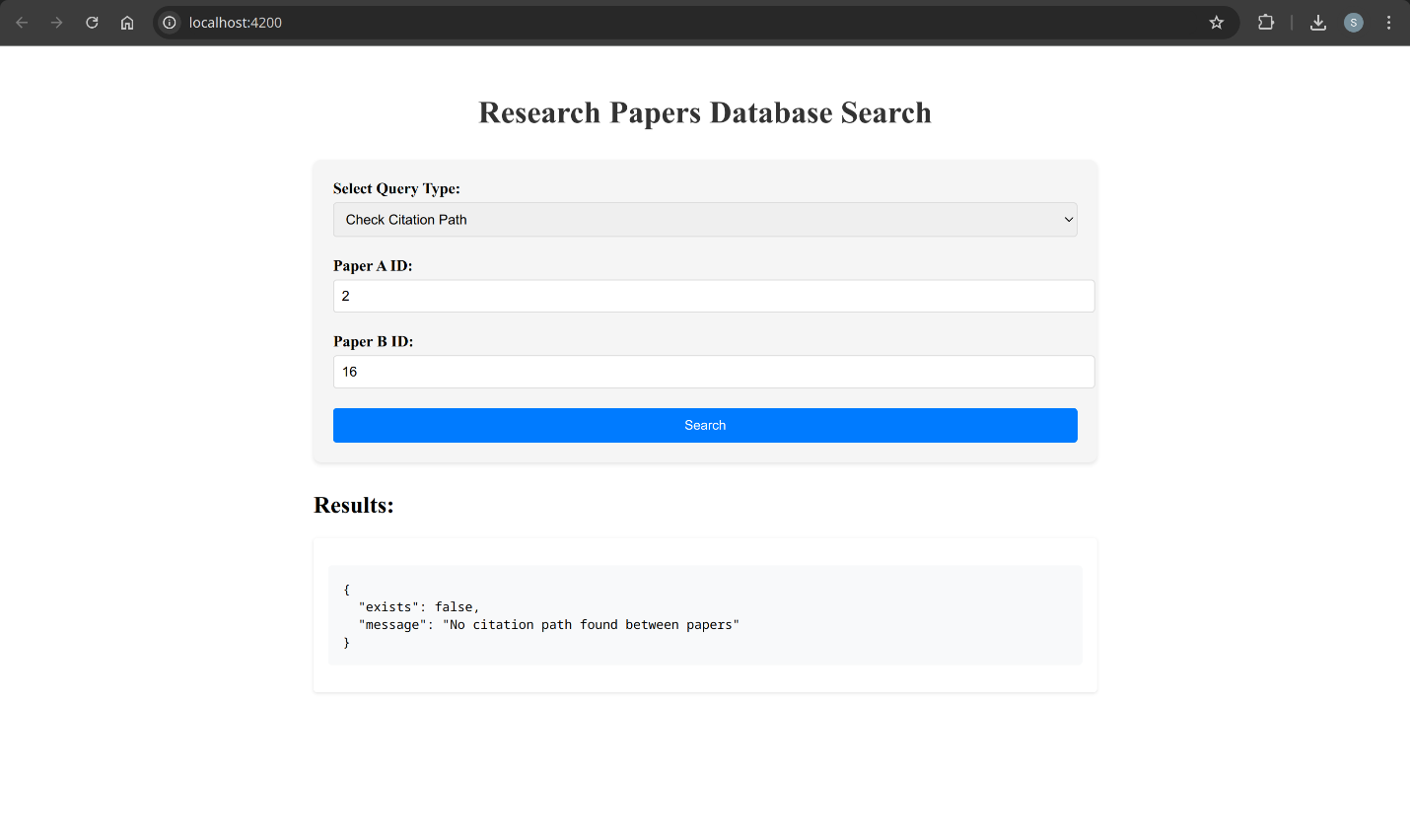
1. **Graph Data Model Concepts**
   * **Nodes**: :Paper, :Author, :Classification
   * **Relationships**:
     + (a:Author)-[:AUTHORED]->(p:Paper)
     + (p1:Paper)-[:CITES]->(p2:Paper)
     + (c:Classification)-[:PARENT\_OF]->(c\_sub:Classification)
     + (p:Paper)-[:HAS\_CLASSIFICATION]->(c:Classification)
2. **Transitive Closure / Path Search**
   * Use Cypher variable‑length patterns (:Paper {id:$A})-[:CITES\*1..]->(:Paper {id:$B}) to detect direct or indirect citation paths.
3. **Hierarchy Retrieval**
   * Ascend the classification tree via [:PARENT\_OF\*0..] to collect the full path.
4. **Indexing & Constraints**
   * Uniqueness constraints on Paper.id, Author.id, Classification.id for fast lookup.

**Conclusion**

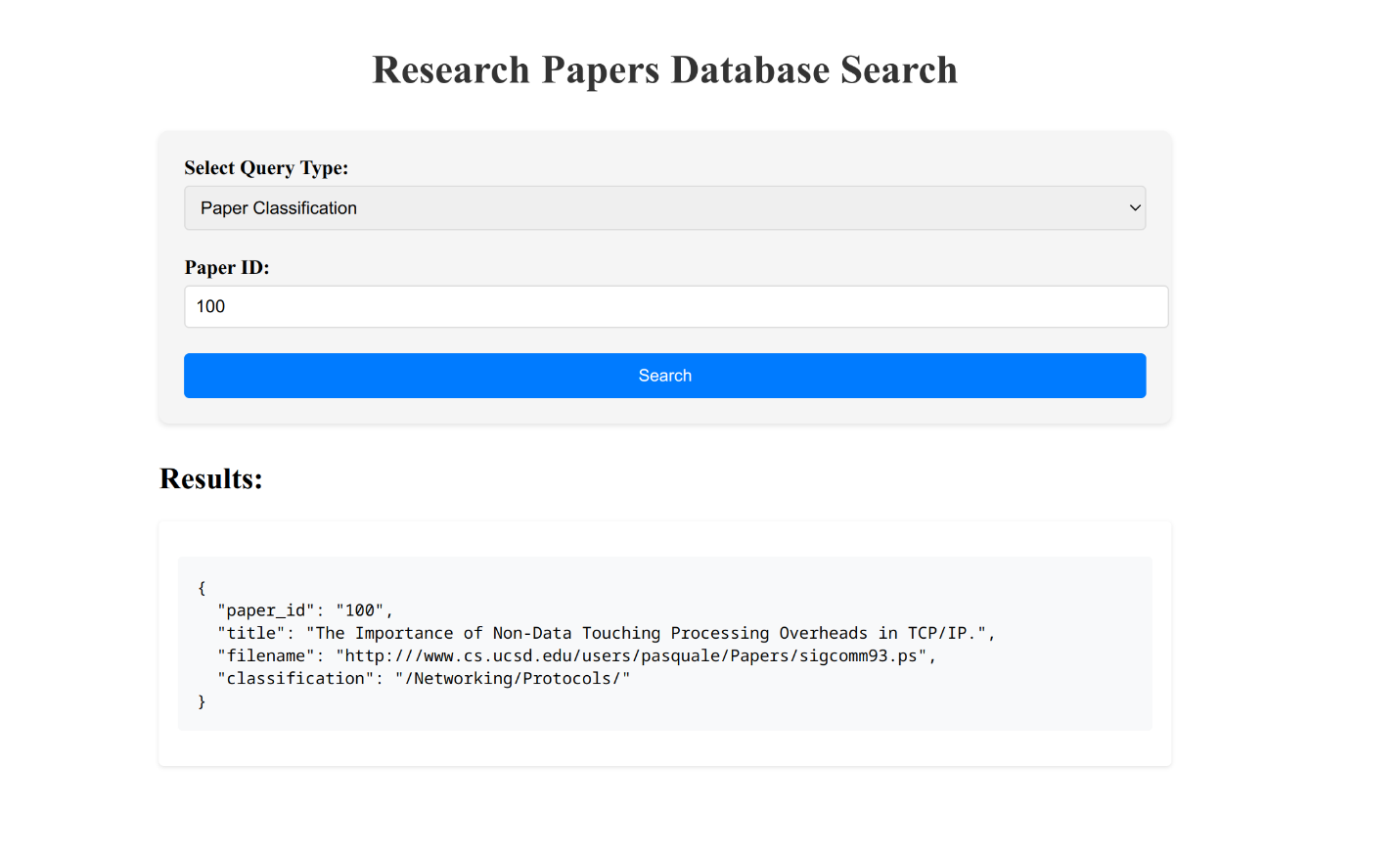
* Successfully modeled the research‑paper domain in Neo4j with authors, citation edges, and a classification hierarchy.
* Imported a large dataset (~282K total elements) and verified graph integrity via counts and sample queries.
* Demonstrated multi‑level citation checks and classification hierarchy retrieval in Neo4j.
* Designed an Angular search interface that interacts with Neo4j over HTTP, supporting the required queries and extensible for custom user queries.

Check citation path checks if the Paper A has citied Paper B

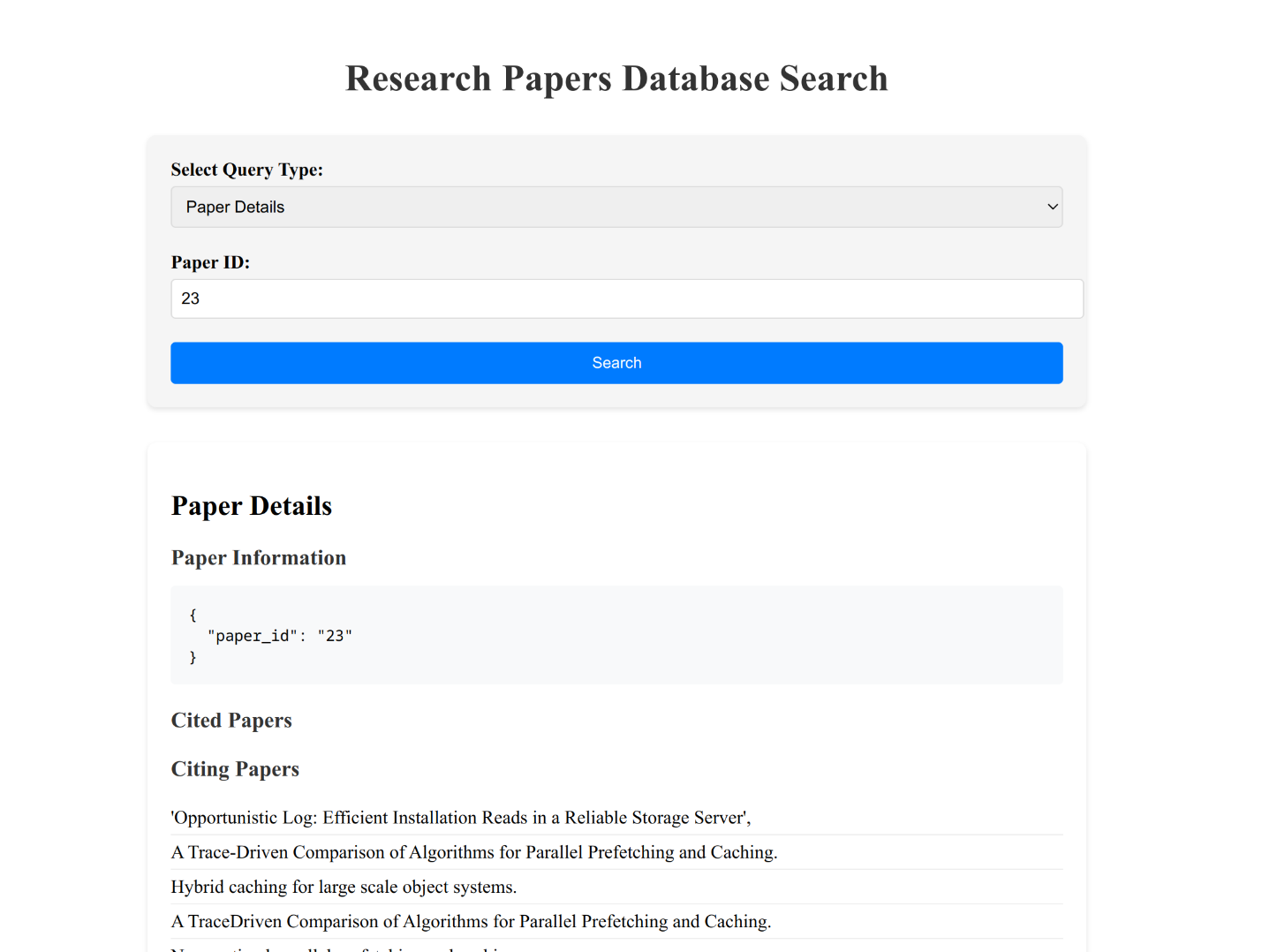




It also gives Under what topic the paper is classified:



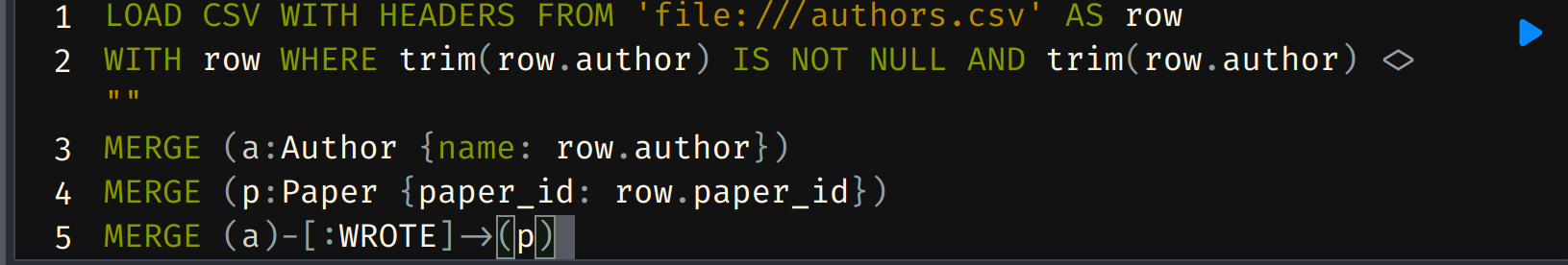
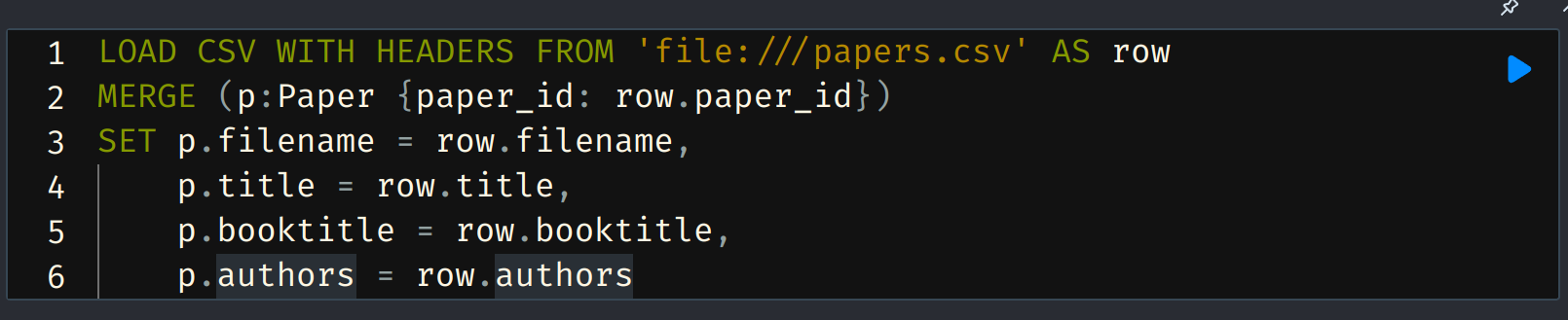
Under Paper details:

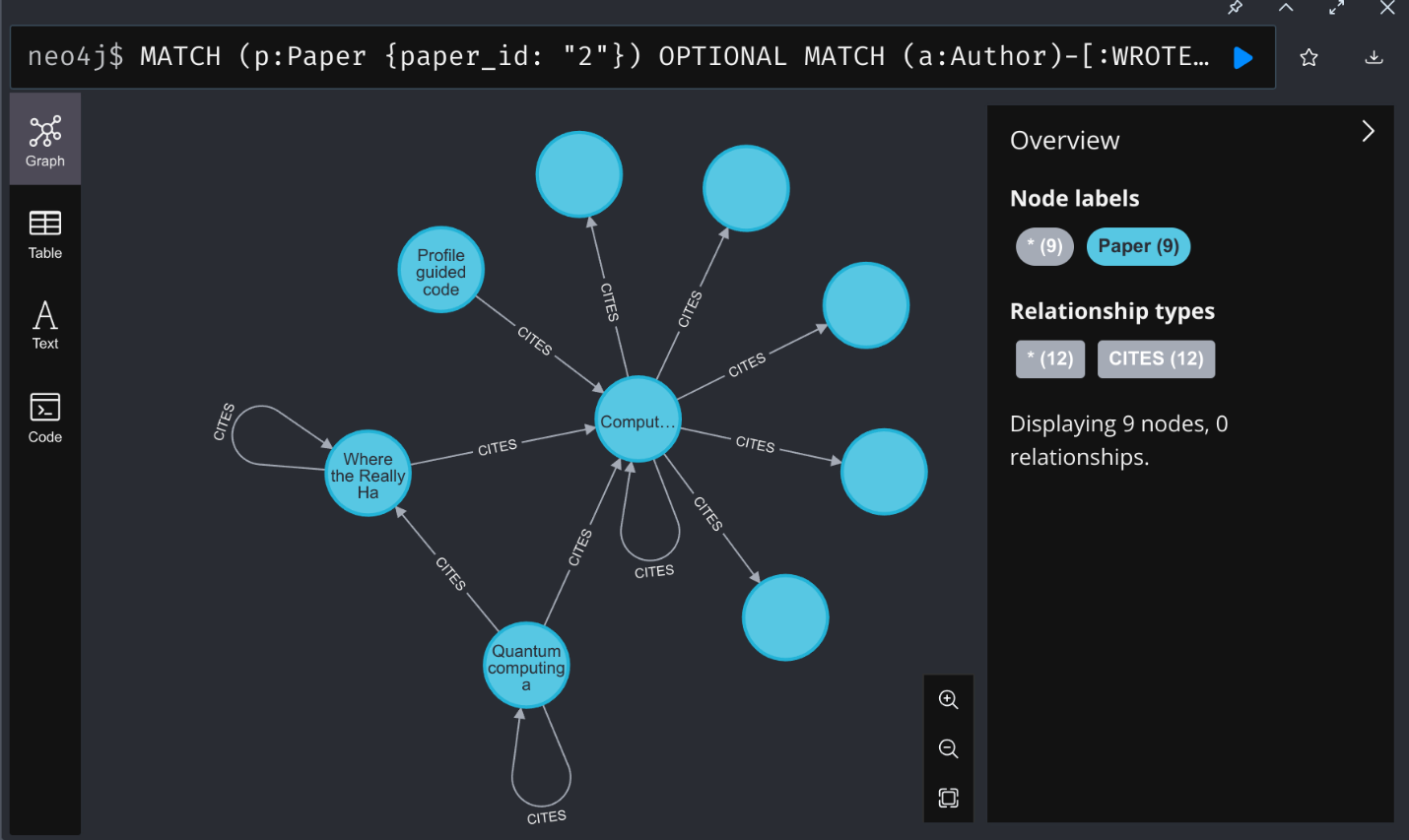
  
And view all papers:



The data was loaded through CSV in neo4j

Running import commands:

  
For each CSV:

Running commands to show citing for paper 2  
  
Running command to show 50 papers

