**Graph Insights using Neo4j and Python**

This project demonstrates how graph databases like Neo4j can be used to model and analyze relationships in educational ecosystems. By leveraging Python and Cypher queries, we extract actionable insights about students, courses, and teachers in a structured yet flexible graph format.

**Technologies Used**

|  |  |
| --- | --- |
| **Tool / Language** | **Purpose** |
| Python | Backend scripting and analysis |
| Neo4j | Graph database engine |
| Cypher | Query language for Neo4j |
| Neo4j Python Driver (neo4j) | Python interface to run Cypher |
| Pandas | Display query results as dataframes |

**Data Model (Graph Schema)**

**Node Types:**

* Student: Represents a student (properties: name, id, etc.)
* Course: Represents a course (properties: name, code, etc.)
* Teacher: Represents a teacher (properties: name, id, etc.)

**Relationship Types:**

* (:Student)-[:ENROLLED\_IN]->(:Course)
* (:Teacher)-[:TEACHES]->(:Course)

**Screenshots:**

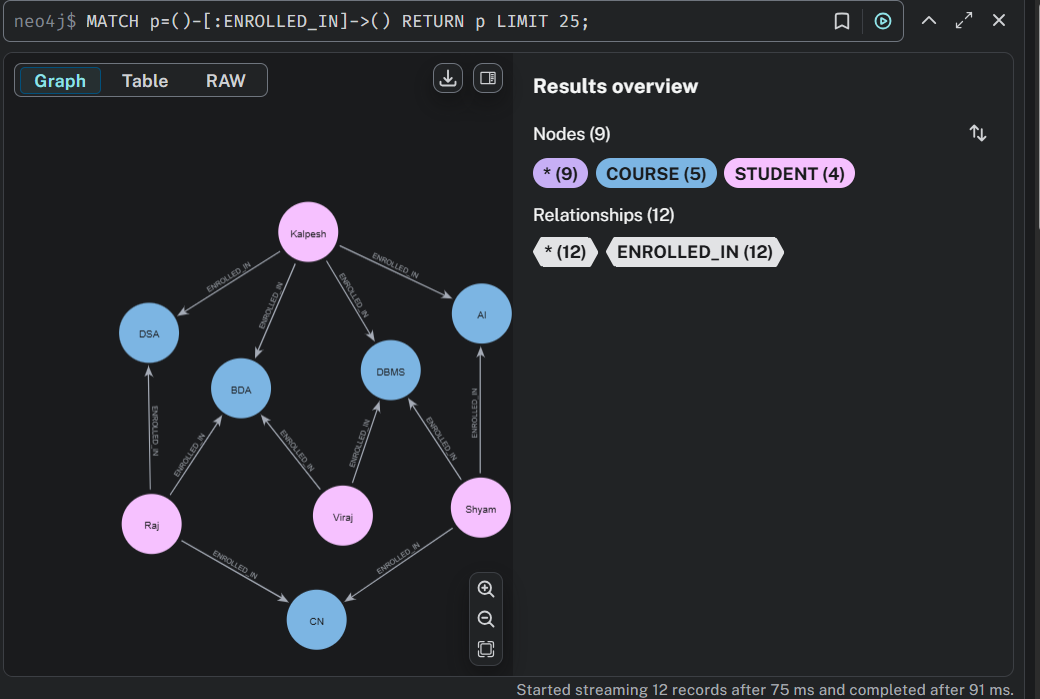
****

Fig: Neo4j Course and student relationship

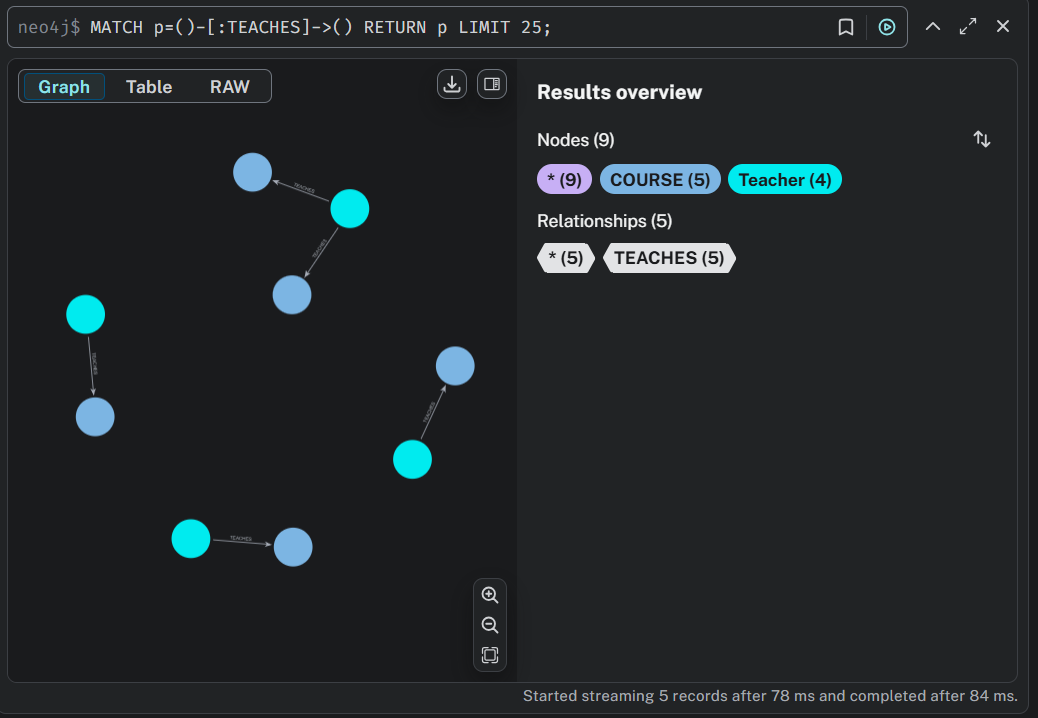
****

Fig: Neo4j Teacher and Course relationship

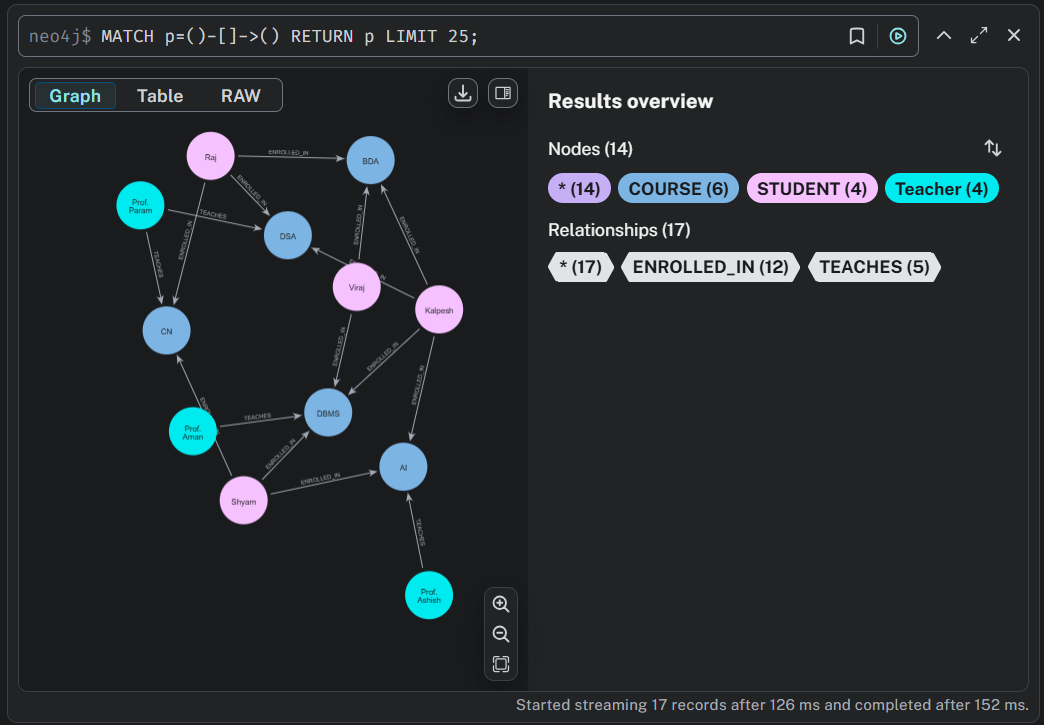
****

Fig: Neo4j full graph

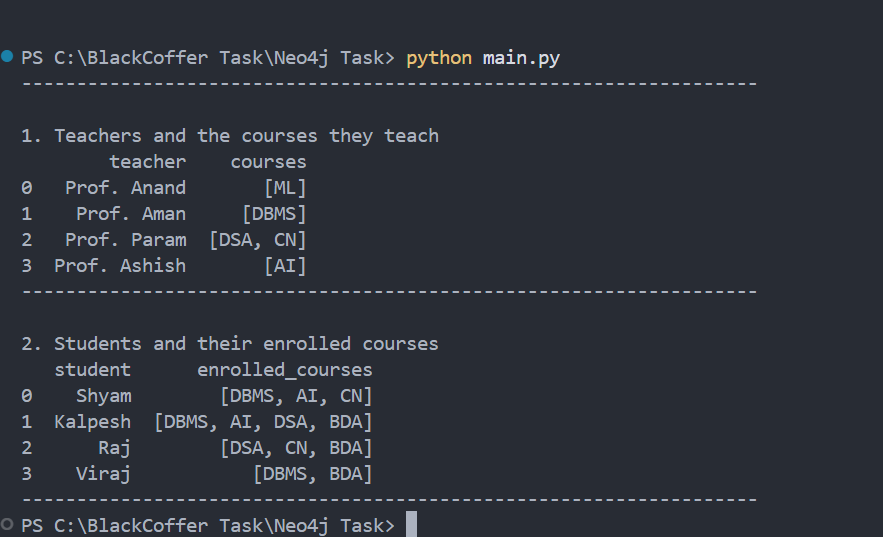
****

Fig: Neo4j using Python