ASSIGNMENT 2

Student ID	Name	Email
8800060	Zarana Gohil	Zgohil0060@conestogac.on.ca
8800149	Basavraj Jaliminche	Bjaliminche0149@econestogac.on.ca
8807575	Dharti Patel	Dpatel757@conestogac.on.ca

Question: Consider the following case:
A student is enrolled into an academic program, which has, among other things, associated schools, and a campus. An academic program has courses that are a part of its curriculum, studied by students and resulting in a grade. Courses have sections in which students are enrolled and taught by instructors associated with a school. Would it be possible for us to represent such a structure using a non-relational definition? Why or why not? Which types of non-relational databases could be used which could not? If a graph can be applied, propose a graph database to support the structure of the above case.

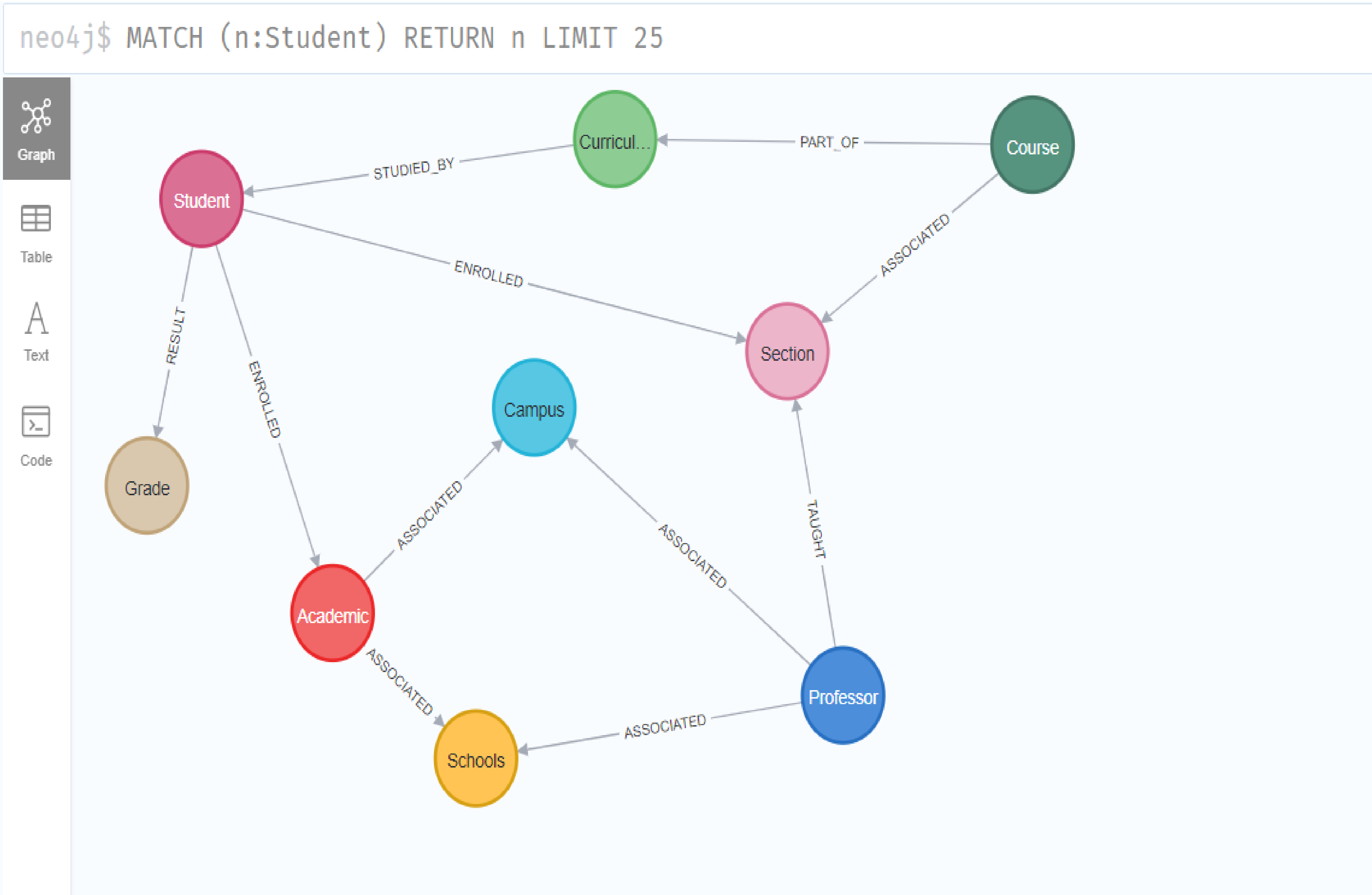
Solution:

Yes, it is possible to represent a given structure using a non-relational database definition. NoSQL database software designed for Web applications, compared to NoSQL database relational database response time is slow when used for large data volume. Among all databases for a given use case, we can use a Graph database. Graph databases are unlike other databases that require you to guess at connections between entities using artificial properties such as foreign keys or out-of-band processing like MapReduce. In the graph model, however, relationships are first-class citizens.

Creating connected structures from simple nodes and their relationships, graph databases enable you to build models that map closely to the problem domain. The resulting models are both simpler and more expressive than those produced using relational databases.

There are three goals generally developer want when they evaluate databases which are

- Intuitiveness: It means a way to create and maintain data in a logical format. This means reducing translation and leads to more productivity.
- Speed: It means the speed of execution and how quickly we can generate results to apply to enable real-time decisions. For example, eBay was able to code their application with this database which made decisions much faster than previous MYSQL
- Agility: It means how easily and quickly code can adapt to business changes. It also provides some constraints that not only maintain data integrity but also provide flexibility to add and remove data easily.


neo4j$ MATCH (n:Student) RETURN n LIMIT 25
Graph
Table
Text
Code
Curricul...
Course
PART_OF
Student
STUDIED_BY
ENROLLED
ASSOCIATED
Section
RESULT
ENROLLED
Campus
Grade
ASSOCIATED
ASSOCIATED
TAUGHT
Academic
ASSOCIATED
Professor
Schools
ASSOCIATED

Cypher code:

```
CREATE (student:Student {title:'Student'})
CREATE (professor:Professor {title:'Professor'})
CREATE (academic_program:Program {title:'Academic Program'})
CREATE (schools:Schools {title:'Schools'})
CREATE (campus:Campus {title:'Campus'})

CREATE (academic_program)-[:ASSOCIATED]->(campus)
CREATE (academic_program)-[:ASSOCIATED]->(schools)

CREATE (student)-[:ENROLLED]->(academic_program)

CREATE (grade:ProgramGrades {title:'Grade'})
CREATE (course:ProgramCourses {title:'Course'})
CREATE (curriculum:ProgramCurriculums {title:'Curriculum'})
CREATE (course)-[:PART_OF]->(curriculum)-[:STUDIED_BY]->(student)
CREATE (student)-[:RESULT]->(grade)

CREATE (section:CourseSections {title:'Section'})
CREATE (course)-[:ASSOCIATED]->(section)
CREATE (student)-[:ENROLLED]->(section)
CREATE (professor)-[:TAUGHT]->(section)

CREATE (professor)-[:ASSOCIATED]->(schools)
CREATE (professor)-[:ASSOCIATED]->(campus)
```