

Bài toán

Bài toán quản lý các khóa học của trường đại học.

Course			
course_id	name	hours	dept_id
CS101	Programming	4	CS
CS201	Algorithms	3	CS
CS202	Systems	3	CS
MA101	Algebra	3	Math
MA201	Calculus	4	Math
MA301	Analysis	4	Math
MU104	Jazz	3	Music
EE102	Circuits	3	EE
IE101	Proabability	3	IE
IE102	Statistics	3	IE

Student		
student_id	name	gpa
11	Bush	3.0
12	Cruz	3.2
13	Clinton	3.9
22	Sanders	3.0
33	Trump	3.8

Department				
dept_id	name	dean	building	room
CS	Computer Science	Rubio	Ajax	100
Math	Mathemagics	Carson	Acme	300
EE	Electrical Engineering	Kasich	Ajax	200
IE	Industrial Engineering	Cruz		200
Music	Musicology	Costello	North	100

Enrollment	
course_id	student_id
CS101	11
MA101	11
CS101	12
CS201	22
MA201	33
EE102	33
MA301	22

Mô hình CSDL quan hệ

student(student_id, name, gpa)

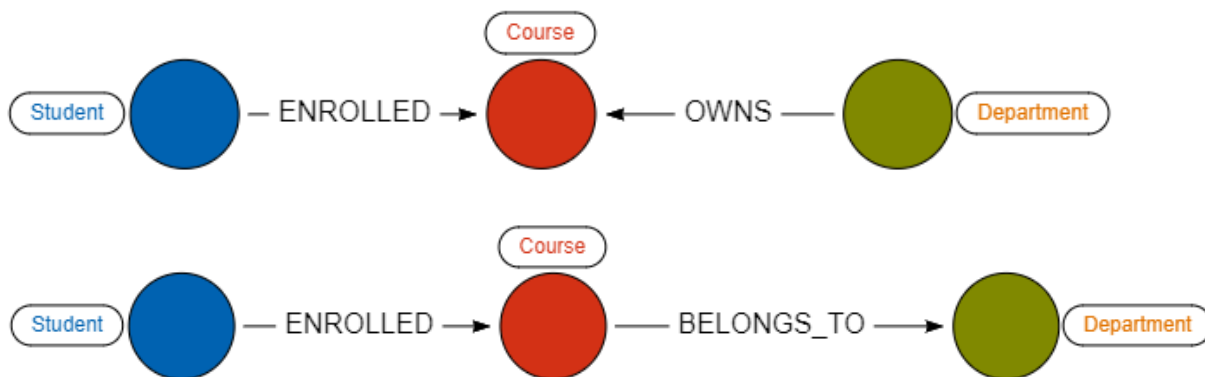
department(dept_id, name, dean, building, room)

course(course_id, name, hours, dept_id) //dept_id là khóa ngoại tham chiếu dept(dept_id)

enrollment(course_id, student_id) //course_id là khóa ngoại tham chiếu course(course_id) và student_id là khóa ngoại tham chiếu student(student_id)

Triển khai mô hình Đồ thị (Graph Model)

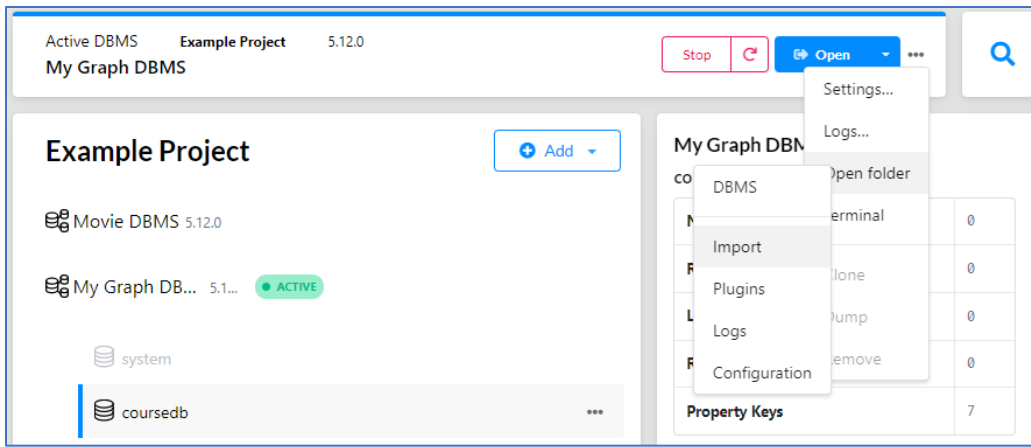
<https://arrows.app>



- Chuyển đổi mô hình dữ liệu quan hệ sang mô hình dữ liệu đồ thị
 - o Một row tương ứng là một node.
 - o Tên table là tương ứng với tên label của node.
 - o Join hoặc khóa ngoại tương ứng là một relationship giữa các node.
- Rows → Nodes, Table names → labels
 - o Mỗi dòng trong bảng student trở thành một node có label là Student trong mô hình đồ thị.
 - o Mỗi dòng trong bảng course trở thành một node có label là Course.
 - o Mỗi dòng trong bảng department trở thành một node có label là Department.
- Joins to relationships
 - o Giữa Student và Course có một relationship tên là ENROLLED (sinh viên đã ghi danh vào khóa học). Mỗi dòng trong bảng enrollment tương ứng thể hiện một relationship.
 - o Giữa Course và Department có một relationship tên là BELONGS_TO (khóa học thuộc về khoa).

Importing the Data using Cypher

- Copy các CSV files vào thư mục import của Neo4j DBMS
`\\Neo4j\Desktop\relate-data\dbmss\dbms-70f621bb-94c3-4eca-8740-cd10f4d5882b\import\courses`
- Viết code Cypher để thực hiện:
 - o Load dữ liệu từ các CSV files
 - o Tạo các indexes hoặc các constraints cho dữ liệu trong đồ thị
 - o Tạo các relationships giữa các nodes



.Neo4jDesktop > relate-data > dbmss > dbms-70f621bb-94c3-4eca-8740-cd10f4d5882b > import > courses				
	Name	Date modified	Type	Size
	courses.csv	18/2/2024 6:19 AM	Microsoft Excel C...	1 KB
	departments.csv	18/2/2024 6:21 AM	Microsoft Excel C...	1 KB
	enrolled.csv	18/2/2024 8:14 AM	Microsoft Excel C...	1 KB
	students.csv	18/2/2024 6:20 AM	Microsoft Excel C...	1 KB

- Có thể cài hai extensions sau: “A Neo4j extension for VS Code” và “Cypher Query Language” trong Visual Studio Code để viết các câu lệnh Cypher code.

<https://neo4j.com/developer-blog/run-cypher-without-leaving-your-ide-with-neo4j-vscode-extension/>

- Add connection
 - o Ctrl + shift + p
 - o Neo4j: Add connection
 - o Scheme: Neo4j
 - o Host: localhost
 - o Port: 7687
 - o Username: neo4j
 - o Password: 12345678
 - o Database: coursedb
- Load dữ liệu từ các CSV files

```
LOAD CSV WITH HEADERS FROM "file:///courses/courses.csv" AS row
MERGE (course:Course { courseID: row.course_id })
SET course.name = row.name, course.hours = toInteger(row.hours)
RETURN course
```

```
LOAD CSV WITH HEADERS FROM "file:///courses/students.csv" AS row
MERGE (student:Student { studentID: row.student_id })
SET student.name = row.name, student.gpa = toFloat(row.gpa)
```

```
LOAD CSV WITH HEADERS FROM "file:///courses/departments.csv" AS row
MERGE (dept:Department { deptID: row.dept_id })
SET dept.name = row.name, dept.dean = row.dean, dept.building = row.building, dept.room = row.room
```

- Tạo các indexes hoặc các constraints cho dữ liệu trong đồ thị


```
CREATE CONSTRAINT unique_course_id FOR (course:Course) REQUIRE course.courseID IS UNIQUE;
```

CREATE CONSTRAINT unique_student_id FOR (student:Student) REQUIRE student.studentID IS UNIQUE;
 CREATE CONSTRAINT unique_dept_id FOR (dept:Department) REQUIRE dept.deptID IS UNIQUE;

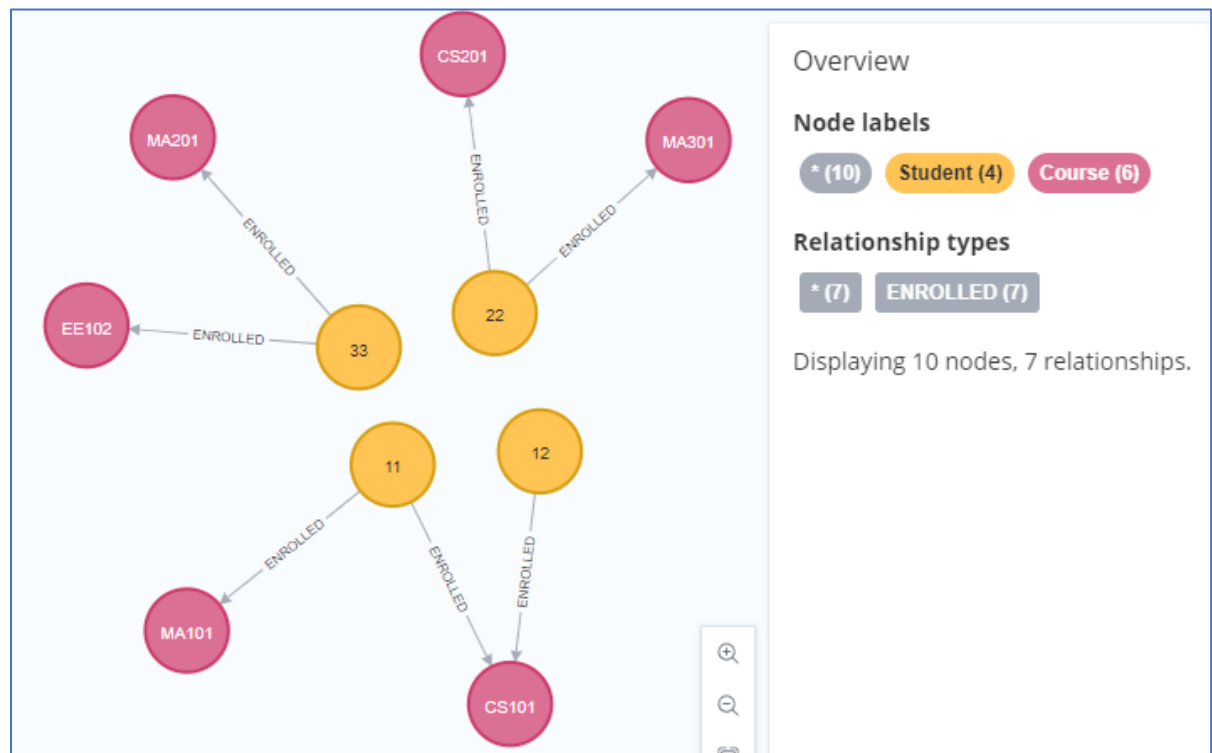
SHOW CONSTRAINTS;

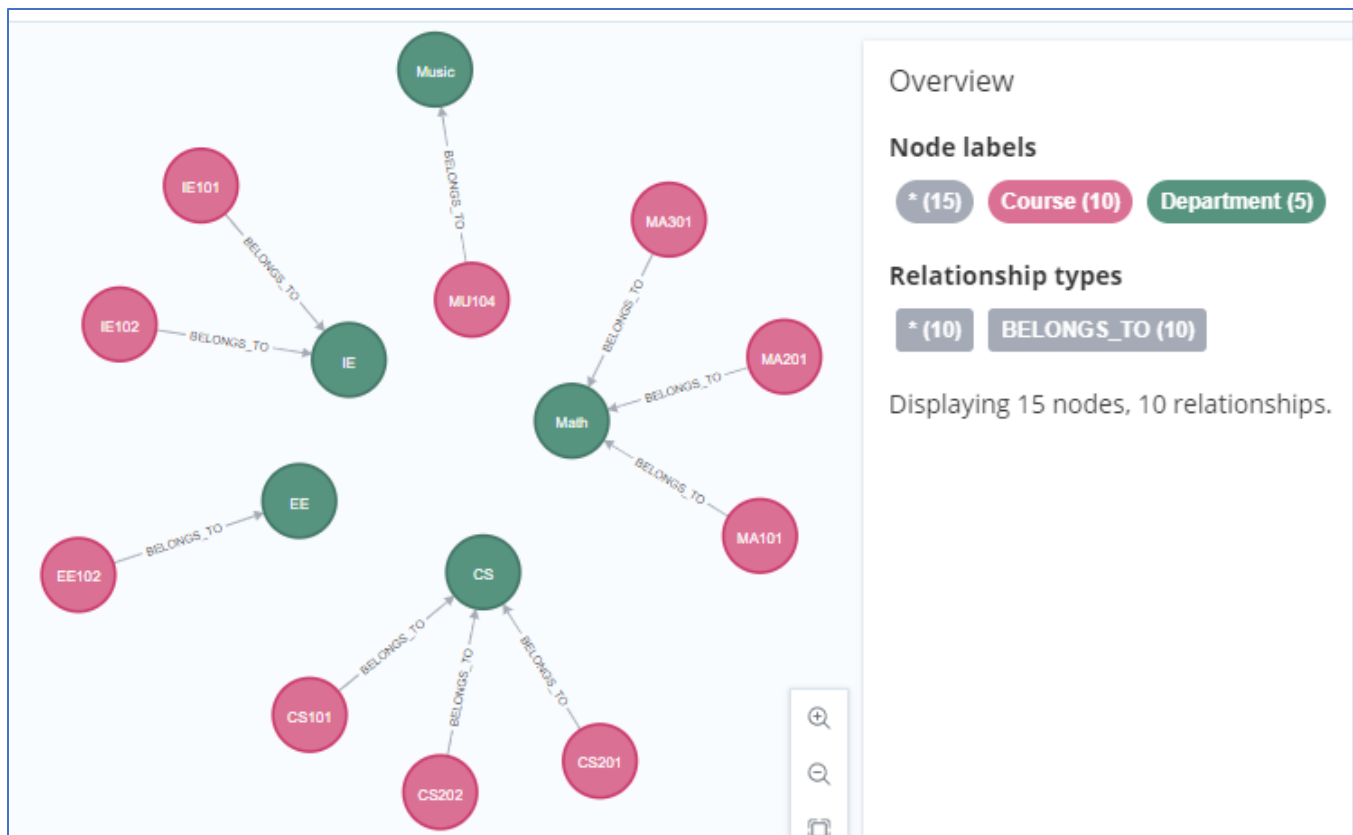
id	name	type	entityType	labelsOrTypes	properties	owned
4	"unique_course_id"	"UNIQUENESS"	"NODE"	["Course"]	["courseID"]	"unique_co"
8	"unique_dept_id"	"UNIQUENESS"	"NODE"	["Department"]	["deptID"]	"unique_de"
6	"unique_student_id"	"UNIQUENESS"	"NODE"	["Student"]	["studentID"]	"unique_st"

- Tạo các relationships giữa các nodes

```
LOAD CSV WITH HEADERS FROM "file:///courses/enrollments.csv" AS row
MATCH (course:Course { courseID: row.course_id })
MATCH (student:Student { studentID:row.student_id })
MERGE (student)-[:ENROLLED]->(course)
```

```
LOAD CSV WITH HEADERS FROM "file:///courses/courses.csv" AS row
MATCH (course:Course { courseID: row.course_id })
MATCH (dept:Department { deptID: row.dept_id })
MERGE (course)-[:BELONGS_TO]->(dept)
```





Querying the Graph

- Liệt kê danh sách n sinh viên

```
MATCH (student:Student)
RETURN student
LIMIT 10;
```
- Tìm kiếm sinh viên khi biết mã số

```
MATCH (student:Student)
WHERE student.studentID = "22"
RETURN student;
```
- Tìm danh sách khóa học thuộc của một khoa nào đó khi biết mã khoa

```
MATCH (course:Course) -[:BELONGS_TO]-> (dept:Department)
WHERE toUpper(dept.deptID) = "MATH"
RETURN course;
```
- Cập nhật name = "Mathematics" cho department_id = "Math"

```
MATCH (dept:Department { deptID: "Math"})
SET dept.name = "Mathematics";

MATCH (dept:Department { deptID: "Math"})
RETURN dept;
```
- Cập nhật name = "Rock n Roll" cho department_id = "Music"

```
MATCH (dept:Department)
WHERE dept.deptID = "Music"
SET dept.name = "Rock n Roll";
```

```
MATCH (dept:Department)
WHERE dept.deptID = "Music"
RETURN dept;
```

6. Thêm khóa học vào khoa IE: IE202, Simulation, 3 hours.

```
CREATE (course:Course { courseID: "IE202", name: "Simulation", hours: 3})
RETURN course;
```

```
MATCH (course:Course { courseID: "IE202"})
RETURN course;
```

7. Xóa toàn bộ các khóa học

Lưu ý, chỉ xóa được những node không có relationship nào.

Hoặc gọi detach để ngắt kết nối và xóa node.

Hoặc xóa relationship trước.

Ví dụ, xóa khóa học có mã số là “IE202” vừa thêm vào thì thành công vì chưa tạo relationship

```
MATCH (course:Course { courseID: "IE202"}) DELETE course
```

Nhưng xóa khóa học có mã số là “CS101” là không thành công vì có sinh viên đã đăng ký tham gia học

```
MATCH (course:Course { courseID: "CS101"}) DELETE course → xóa không thành công
```

Detach để ngắt kết nối và xóa

```
MATCH (course:Course { courseID: "CS101"}) DETACH DELETE course → xóa thành công
```

```
MATCH (n: Course) detach delete n → xóa hết
```

Hoặc xóa relationship trước

Xóa các relationship

```
MATCH(() - [r:ENROLLED]->()) DELETE r;
```

```
MATCH(() - [r:BELONGS_TO]->()) DELETE r;
```

Xóa tất cả các node có nhãn là Course

```
MATCH (course:Course) DELETE course;
```

```
MATCH (course:Course) RETURN course;
```

8. Liệt kê tất cả các khoa

9. Liệt kê tên của tất cả các trưởng khoa

```
MATCH (dept:Department) RETURN dept.dean
```

10. Tìm tên của trưởng khoa CS

11. Liệt kê tất cả các khóa học của CS và IE

12. Liệt kê danh sách các tên của các sinh viên đăng ký học khóa học CS101

```
MATCH (student:Student)-[:ENROLLED]->(course:Course)
```

```
WHERE course.courseID = "CS101"
```

```
RETURN student.name
```

13. Tổng số sinh viên đăng ký học của mỗi khoa

14. Tổng số sinh viên đăng ký học của mỗi khoa, kết quả sắp xếp theo mã khoa

15. Tổng số sinh viên đăng ký học của mỗi khoa, kết quả sắp xếp theo số sinh viên

16. Liệt kê danh sách tên của các trưởng khoa mà các khoa này không có sinh viên đăng ký học

17. **Danh sách** khoa có số sinh viên đăng ký học nhiều nhất

18. Danh sách sinh viên có gpa ≥ 3.2 , kết quả sắp xếp giảm dần theo gpa

Neo4j for Java Developers

<https://neo4j.com/developer/java/>

```
<dependency>  
  <groupId>org.neo4j.driver</groupId>  
  <artifactId>neo4j-java-driver</artifactId>  
  <version>5.17.0</version>  
</dependency>
```

Neo4j Developer Graph Apps Gallery

<https://install.graphapp.io/>

Neo4j Developer Graph Apps Gallery - One-Click Install of Curated Graph Developer Tools

Neo4j Desktop

Neo4j Desktop is a control center to manage your local and remote Neo4j graphs in one place. You can [download and install](#) it, and [learn more about it here](#).

Graph Apps

The apps below are additions to Neo4j Desktop that provide new capabilities, like Monitoring, Import, Analysis, Running Graph Algorithms, Visualization and much more

Neo4j ETL Tool Neo4j Labs



Import data from relational databases
into Neo4j



Install

Graph Data Science Playground Neo4j Labs



Playground for Neo4j Graph
Algorithms



Install

GraphQL Toolbox Neo4j, Inc.



Build, query, and deploy GraphQL APIs
backed by Neo4j.



Install

Graph Gallery Neo4j Labs



Graph Examples on your desktop.



Install

Charts Neo4j Labs

APOC User Guide Neo4j Labs

Neosemantics Neo4j Labs

NeoDash Neo4j Labs

Neo4j ETL Tool

Import data from relational databases into neo4j

JDBC Connection

Database access requires an authenticated connection and a valid JDBC driver.

Connection Name

Host

localhost

Port

3306

Type

mysql

Database

qlsinhvien

Connection URL

jdbc:mysql://localhost:3306/qlsinhvien?autoReconnect=true&useSSL=false&useCursorFetch=true&allowPublicKeyRetrieval=true

Username

root

Password

....

BACK TO START


TEST AND SAVE CONNECTION

Load Data Model

Select the project: Example Project

Select what connection you want to import from or add a new one.


Selected RDBMS instance **NONE**

 Database: QLSINHVIENType: MYSQL

Select what Neo4j instance you want to import to

Selected Neo4j instance **My Graph DBMS**

 Movie DBMS
STOPPED
5.12.0 - enterprise

 My Graph DBMS
RUNNING
5.12.0 - enterprise

ADD CONNECTION

SEE LOGS

START MAPPING

NEXT

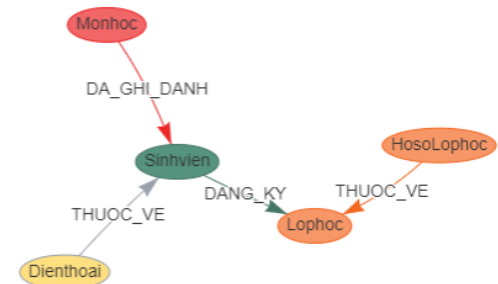
Explore and change your metadata

NODES

RELATIONSHIPS

Search by entity name

Entity Name	Actions	Skip
<div>●</div> <div>Monhoc</div>	<div></div>	<div></div>
<div>●</div> <div>Lophoc</div>	<div></div>	<div></div>
<div>●</div> <div>HosoLophoc</div>	<div></div>	<div></div>
<div>●</div> <div>Dienthoai</div>	<div></div>	<div></div>
<div>●</div> <div>Sinhvien</div>	<div></div>	<div></div>



BACK TO START

SAVE MAPPING

NEXT

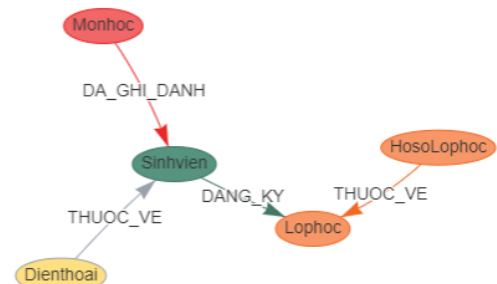
Explore and change your metadata

NODES

RELATIONSHIPS

Search by entity name

Entity Name	Start Node	End Node	Actions	Skip
THUOC_VE	HosoLophoc	Lophoc		<input type="checkbox"/>
THUOC_VE	Dienthoai	Sinhvien		<input type="checkbox"/>
DANG_KY	Sinhvien	Lophoc		<input type="checkbox"/>
DA_GHI_DANH	Monhoc	Sinhvien		<input type="checkbox"/>



BACK TO START

SAVE MAPPING

NEXT

Import your data into Neo4j

Press the button and start import your data

FROM

Connection name:
From database: *QLSINHVIEN*
Of type: *MYSQL*



TO

Instance name: *My Graph DBMS*
Version: *5.12.0 - enterprise (LOCAL)*
Status: *RUNNING*

Import Mode

Online direct import From SQL

The import mode type

Neo4j Database

qlsinhvien

Select the target database

Unwind Row Size *

1000

Number of rows processed per UNWIND

Transaction Batch Size *

10000

Number of rows processed per Transaction

BACK TO START

BACK TO MAPPING

IMPORT DATA

qlsinhvien

Node labels

*{5} Dienthoai HosoLophoc
Lophoc Monhoc Sinhvien

Relationship types

*{3} DA_GHI_DANH THUOC_VE

Property keys

diemthi gioitinh ho
hockygd lanthi mota mshs
mslop msmh mssv
ngaylap ngaysinh sisoDukien
soDienthoai sotinchi ten
tenlop tenmh

qlsinhvien\$ MATCH p=()-[r:DA_GHI_DANH]->() RETURN p LIMIT 25

