

BUILDING A KNOWLEDGE GRAPH ON NEO4J TO SUPPORT KNOWLEDGE RETRIEVAL FOR DATABASE COURSE

Luong Tran Hy Hien*, Tran Quoc Khanh, Ha Thai Khanh, Truong Anh Kiet, Nguyen Do Thanh Phu
Ho Chi Minh City University of Education

ARTICLE INFO		ABSTRACT
Received:	12/3/2026	This paper proposes building a Knowledge Graph on the Neo4j platform to overcome the limitations of traditional retrieval methods (F1-score 57%) in performing multi-hop reasoning for the Database Fundamentals course. Entities and relationships were extracted from Vietnamese textbooks using rule-based pattern-matching rules, resulting in a knowledge graph composed of 70 concept nodes and 150 relational edges. The retrieval mechanism processes natural language into Cypher queries. An experimental evaluation on a set of 50 benchmark questions demonstrates that the system achieves an F1-score of 92% (an improvement of 35% compared to keyword-based lookup methods), an average response time of 150 milliseconds, and 88% accuracy for multi-hop inference. This research contributes a standardized Vietnamese Knowledge Graph dataset and a scalable multidimensional retrieval model for engineering education. These results indicate that integrating structured domain knowledge with graph-based retrieval and natural language-to-Cypher translation substantially enhances both the precision and reasoning capabilities of academic information access systems.
Revised:	28/5/2026	
Published:	29/5/2026	
KEYWORDS		
Knowledge Graph		
Knowledge retrieval		
Database		
Digital education		
Neo4J		

XÂY DỰNG ĐỒ THỊ TRI THỨC TRÊN NEO4J HỖ TRỢ TRA CỨU KIẾN THỨC MÔN HỌC CƠ SỞ DỮ LIỆU

Lương Trần Hy Hiến*, Trần Quốc Khánh, Hà Thái Khánh, Trương Anh Kiệt, Nguyễn Đỗ Thanh Phú
Trường Đại học Sư phạm Thành phố Hồ Chí Minh

THÔNG TIN BÀI BÁO		TÓM TẮT
Ngày nhận bài:	12/3/2026	Bài báo đề xuất xây dựng đồ thị tri thức trên nền tảng Neo4j nhằm khắc phục hạn chế suy luận đa bước của các phương pháp tra cứu truyền thống (F1-score 57%) cho môn học Cơ sở dữ liệu. Các thực thể và mối quan hệ được trích xuất từ giáo trình tiếng Việt bằng cách sử dụng các quy tắc so khớp mẫu dựa trên luật, hình thành một đồ thị gồm 70 nút khái niệm và 150 cạnh quan hệ. Cơ chế tra cứu được thực hiện qua việc chuyển đổi ngôn ngữ tự nhiên thành truy vấn Cypher. Đánh giá thực nghiệm trên một tập hợp 50 câu hỏi chuẩn cho thấy hệ thống đạt điểm F1-score là 92% (tăng 35% so với các phương pháp tra cứu dựa trên từ khóa), thời gian phản hồi trung bình 150 ms và độ chính xác suy luận đa bước đạt 88%. Nghiên cứu đóng góp bộ dữ liệu đồ thị tri thức tiếng Việt chuẩn hóa và một mô hình tra cứu có thể mở rộng cho các lĩnh vực kỹ thuật. Những kết quả này cho thấy việc tích hợp kiến thức miền có cấu trúc với truy xuất dựa trên đồ thị và chuyển đổi truy vấn ngôn ngữ tự nhiên sang Cypher đã cải thiện đáng kể cả độ chính xác và khả năng suy luận của hệ thống truy xuất thông tin học thuật.
Ngày hoàn thiện:	28/5/2026	
Ngày đăng:	29/5/2026	
TỪ KHÓA		
Đồ thị tri thức		
Tra cứu kiến thức		
Cơ sở dữ liệu		
Giáo dục số		
Neo4J		

DOI: <https://doi.org/10.34238/tnu-jst.15000>

* Corresponding author. Email: hienlth@hcmue.edu.vn

1. Giới thiệu

1.1. Bối cảnh và động lực

Môn học Cơ sở dữ liệu (CSDL) là nền tảng Công nghệ thông tin, có cấu trúc kiến thức phức tạp với hơn 70 khái niệm từ mô hình dữ liệu đến tối ưu hóa truy vấn và hàng trăm mối quan hệ ngữ nghĩa giữa chúng [1]. Sinh viên thường gặp khó khăn trong việc nắm bắt mối liên hệ giữa các khái niệm như “Khóa ngoại”, “JOIN”, “Indexing” do tính trừu tượng và độ sâu của kiến thức [2].

Các phương pháp tra cứu kiến thức truyền thống (keyword search) dựa trên so khớp chuỗi ký tự thiếu khả năng suy luận đa cấp (multi-hop reasoning) và không khám phá được kiến thức liên kết ẩn [3], [4]. Ví dụ, khi sinh viên tìm kiếm “SQL hỗ trợ truy vấn nào?”, phương pháp truyền thống không thể trả về các khái niệm liên quan như “SELECT”, “JOIN”, “Subquery” cùng với mối quan hệ ngữ nghĩa của chúng.

1.2. Đồ thị tri thức – Giải pháp cho giáo dục

Đồ thị tri thức (Knowledge Graph – KG) là mô hình biểu diễn tri thức dưới dạng đồ thị có hướng $G = (V, E, \lambda)$, trong đó V là tập nút (entities/concepts), E là tập cạnh (relationships) có hướng, và $\lambda: V \cup E \rightarrow \Theta$ là ánh xạ thuộc tính (properties) [5], [6]. KG cho phép mã hóa tri thức của một miền dưới dạng mạng lưới, giúp máy tính có thể thực hiện các thao tác truy vấn, suy luận, lan truyền thông tin một cách tự nhiên theo cấu trúc đồ thị. Mặc dù trên thế giới đã có những công trình đạt độ chính xác cao như CourseKG [2] hay EDUKG [3], các KG này chủ yếu hỗ trợ tiếng Anh hoặc tiếng Trung. Tại Việt Nam, các nghiên cứu gần đây (Nguyễn Việt Hưng và cộng sự [7]; Trần Khải Thiện và cộng sự [8]) vẫn chưa tập trung xây dựng một KG chuẩn hóa bằng tiếng Việt, đặc biệt là cho các môn học chuyên ngành như Cơ sở dữ liệu, cũng chưa hỗ trợ tra cứu tự nhiên với multi-hop reasoning [9].

Bên cạnh rào cản ngôn ngữ, việc ứng dụng KG trong thực tế giáo dục còn gặp khó khăn do người dùng phải nắm vững các ngôn ngữ truy vấn phức tạp (như Cypher, SPARQL) để khai thác thông tin [10]. Hơn nữa, nhiều nghiên cứu hiện tại vẫn còn thiếu các đánh giá định lượng bài bản so với các phương pháp tra cứu truyền thống [11], [12].

Để giải quyết các thách thức trên, nghiên cứu này ứng dụng Neo4j – một hệ quản trị cơ sở dữ liệu đồ thị linh hoạt và trực quan – để mô hình hóa các khái niệm học thuật của môn CSDL. Từ đó, bài báo đề xuất một hệ thống truy xuất thông minh, cho phép sinh viên tra cứu trực tiếp bằng ngôn ngữ tự nhiên tiếng Việt mà không cần kiến thức về ngôn ngữ truy vấn đồ thị.

Mục tiêu và những đóng góp chính của bài báo được tóm tắt như sau:

1. Xây dựng Đồ thị tri thức chuyên dụng: Đề xuất quy trình thực tế và trực tiếp triển khai một KG tiếng Việt (gồm 70 khái niệm, 150 quan hệ) dành riêng cho học phần Cơ sở dữ liệu trên nền tảng Neo4j.

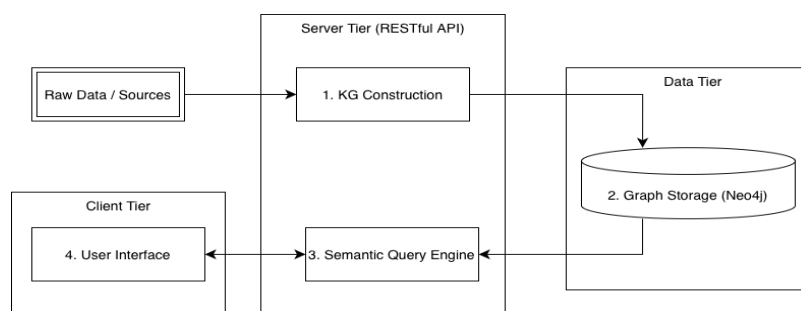
2. Phát triển hệ thống hỏi–đáp ngữ nghĩa: Thiết kế kiến trúc hệ thống tự động phân tích câu hỏi tiếng Việt tự nhiên, sinh truy vấn Cypher và trả lời dựa trên KG.

3. Đánh giá thực nghiệm và công bố dữ liệu: Thực hiện đánh giá định lượng hệ thống trên bộ 50 câu hỏi chuẩn, so sánh hiệu năng với phương pháp tra cứu từ khóa truyền thống. Đồng thời, nghiên cứu cung cấp bộ dữ liệu KG và tập hỏi–đáp chuẩn có thể tái sử dụng cho các nghiên cứu tiếp theo.

2. Phương pháp đề xuất

2.1. Tổng quan kiến trúc hệ thống

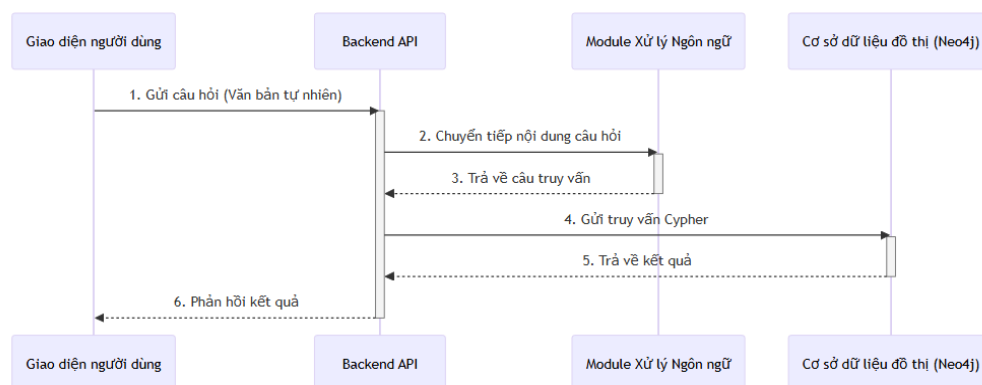
Hệ thống gồm bốn module chính: (1) KG Construction, (2) Graph Storage (Neo4j), (3) Semantic Query Engine và (4) User Interface (UI), tổ chức theo mô hình client–server với RESTful API như minh họa trong Hình 1.



Hình 1. Kiến trúc tổng thể hệ thống tra cứu kiến thức dựa trên KG

Luồng xử lý tổng quát như sau: sinh viên nhập câu hỏi tiếng Việt trên Web UI; frontend gửi yêu cầu POST đến backend (Node.js/Express); Semantic Query Engine phân tích câu hỏi và sinh truy vấn Cypher; Neo4j thực thi truy vấn trên đồ thị tri thức; kết quả được sắp xếp theo mức độ liên quan, trả về dạng JSON và hiển thị trực quan cho người dùng.

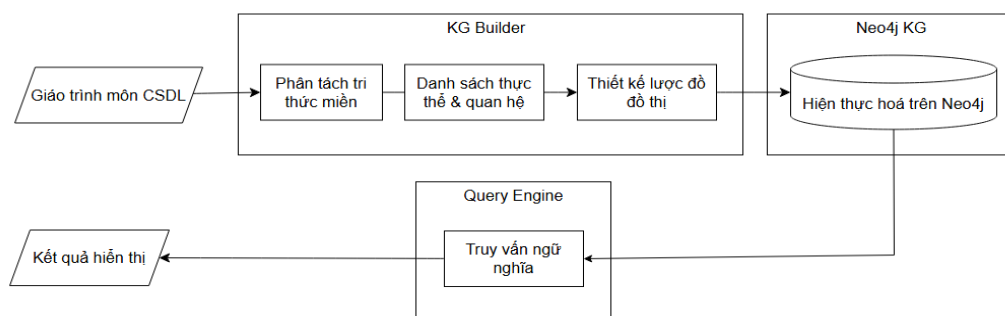
Ứng dụng được triển khai theo kiến trúc web ba lớp như Hình 2. Bao gồm: (i) lớp giao diện người dùng cho phép nhập câu hỏi và xem kết quả; (ii) lớp dịch vụ ứng dụng hiện thực API /api/search để xử lý ngôn ngữ và tương tác với cơ sở dữ liệu đồ thị; (iii) lớp dữ liệu Neo4j lưu trữ toàn bộ KG và xử lý các truy vấn Cypher.



Hình 2. Kiến trúc và luồng xử lý của hệ thống tra cứu

2.2. Xây dựng đồ thị tri thức cho môn cơ sở dữ liệu

Quy trình xây dựng KG cho môn Cơ sở dữ liệu được tổ chức thành bốn giai đoạn liên tiếp được mô tả ở Hình 3: (1) phân tích tri thức miền, (2) thiết kế lược đồ đồ thị, (3) hiện thực hóa trên Neo4j và (4) chuẩn bị cho tầng truy vấn ngữ nghĩa.



Hình 3. Quy trình xây dựng KG cho môn CSDL

2.2.1. Phân tích tri thức miền

Nhóm nghiên cứu thu thập đề cương, giáo trình, slide và đề thi/bài tập của học phần để hình thành danh sách khái niệm ứng viên. Từ các tài liệu này, 70 khái niệm chính được xác định và phân loại theo các nhóm: Concept (CSDL, Bảng, Khóa chính, Khóa ngoại,...), DataModel (Relational, ER, NoSQL,...), Query (SQL, SELECT, JOIN,...), Optimization (Index, Query Plan,...) và Transaction (ACID, Lock, Recovery,...).

Từ câu văn trong giáo trình, khoảng 150 quan hệ được trích xuất bằng pattern matching, gom vào 8 loại quan hệ chính: QUAN_HỆ, LÀ_MỘT_LOẠI, CÓ_THUỘC_TÍNH, HỖ_TRỢ, YÊU_CẦU, SỬ_DỤNG, TỐI_ƯU_BẢNG, LIÊN QUAN. Với mỗi khái niệm, hệ thống lưu các thuộc tính: id (N1...N70), name (tên tiếng Việt), definition (1–3 câu định nghĩa), example (ví dụ schema hoặc câu lệnh SQL) và category (nhãn phân loại như Concept, DataModel, Query,...).

2.2.2. Thiết kế lược đồ đồ thị

Trên tập khái niệm ứng viên, nhóm xác định tập khái niệm cốt lõi khoảng 70 phần tử và các kiểu quan hệ ngữ nghĩa chủ đạo như BAO_GOM, QUAN_LY, HO_TRO, DAM_BAO, LIEN_KET_VOI.

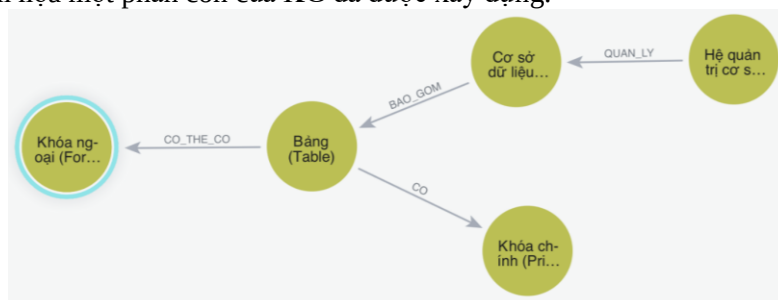
Đồ thị tri thức được mô hình hóa dưới dạng $KG=(V,E)$, trong đó:

$$\mathcal{KG}_{CSDL} = (V, E, \lambda) \#(1)$$

với:

- $|V| = 70$ (số concepts)
- $|E| = 150$ (số relationships)
- $|R| = 8$ (số loại quan hệ)
- $|\Theta| = 5$ (số properties: id, name, definition, example, category)

Hình 4 minh họa một phần con của KG đã được xây dựng.



Hình 4. Minh họa một phần con của KG

2.2.3. Triển khai KG trên Neo4J

Sau khi xác định tập khái niệm (đỉnh) và quan hệ (cạnh), KG được hiện thực hóa trên hệ quản trị cơ sở dữ liệu đồ thị Neo4j thông qua các câu lệnh Cypher. Quá trình triển khai bao gồm các bước chính sau:

a. Khởi tạo dữ liệu (Schema Design & Import)

- Node Concept: Thuộc tính id (N1–N70), name, definition, example, category.
- Relationship: 8 loại nhãn QUAN_HỆ, LÀ_MỘT_LOẠI, CÓ_THUỘC_TÍNH, HỖ_TRỢ, YÊU_CẦU, SỬ_DỤNG, TỐI_ƯU_BẢNG, LIÊN QUAN.

Đoạn mã Cypher mẫu mô tả cấu trúc:

```
// Tạo Node
CREATE (n:Concept {id: 'N1', name: 'Cơ sở dữ liệu', definition: '...', example: '...', category: 'Concept'})
// Tạo Relationship
MATCH (a:Concept {id: 'N1'}), (b:Concept {id: 'N2'}) CREATE (a)-[:QUAN_HỆ]->(b)
```

b. Tối ưu hóa hiệu năng (Performance Tuning)

- Chỉ mục (Indexing): Tạo B-tree index trên các thuộc tính name và id của nhân Concept để tăng tốc độ tìm kiếm. Ví dụ: `CREATE INDEX concept_name FOR (c:Concept) ON (c.name)`.
- Cấu hình bộ nhớ: Phân bổ tài nguyên phù hợp với tập dữ liệu thông qua cấu hình `dbms.memory.heap.max_size=2G` và `dbms.memory.pagecache.size=1G`.
- Tối ưu truy vấn: Sử dụng công cụ EXPLAIN và PROFILE để đánh giá và hiệu chỉnh các câu lệnh Cypher.

2.3. Hệ thống tra cứu kiến thức môn học

2.3.1. Kiến trúc hệ thống

Hệ thống tra cứu kiến thức môn học cơ sở dữ liệu được xây dựng nền web, sử dụng:

- Frontend: Giao diện tương tác dựa trên EJS template engine.
- Backend: Xử lý logic và điều phối truy vấn bằng Node.js (Express.js).
- Database: Lưu trữ và quản lý tri thức đồ thị trên Neo4j.

2.3.2. Quy trình xử lý truy vấn ngữ nghĩa (pipeline)

Thay vì yêu cầu người dùng sử dụng ngôn ngữ Cypher, hệ thống triển khai một pipeline tự động chuyển đổi ngôn ngữ tự nhiên thành truy vấn đồ thị qua 4 giai đoạn chính:

Giai đoạn 1: Khởi tạo từ điển tri thức

Hệ thống trích xuất toàn bộ danh sách Concept từ Neo4j để xây dựng bộ từ điển (conceptDictionary) và bản đồ ánh xạ (conceptMap). Tên khái niệm được tách thành hai biến thể tiếng Việt và tiếng Anh (nếu có), sắp xếp theo độ dài giảm dần. Để tăng độ chịu lỗi, conceptDictionary được mở rộng với các biến thể từ đồng nghĩa (ví dụ: “bảng” ↔ “table”, “quan hệ” ↔ “relation”) và viết tắt thông dụng (ví dụ: “CSDL”, “FK”, “PK”) được bổ sung thủ công dựa trên khảo sát cách dùng của sinh viên.

Giai đoạn 2: Tách thực thể bằng thuật toán so khớp tối đa hai chiều (Bi-MM)

Để nhận diện chính xác các khái niệm trong câu hỏi tiếng Việt (vốn có cấu trúc phức tạp), hệ thống thực hiện:

- Chuẩn hóa: Chuyển câu hỏi về dạng chữ thường và loại bỏ khoảng trắng dư thừa.
- So khớp xuôi (Forward Maximum Matching): Tách từ từ trái sang phải.
- So khớp ngược (Backward Maximum Matching): Tách từ từ phải sang trái.
- Tối ưu hóa: So sánh kết quả của hai hướng và chọn tập hợp có số lượng phần tử ít hơn để hạn chế tình trạng phân mảnh ngữ nghĩa (ví dụ: ưu tiên cụm “truy vấn dữ liệu” thay vì tách rời thành “truy vấn” và “dữ liệu”).

Lưu ý: Bi-MM phụ thuộc vào conceptDictionary; hệ thống sẽ chuyển sang chế độ keyword fallback nếu đầu vào chứa từ không có trong từ điển (ví dụ: từ lóng hoặc lỗi chính tả ngoài tập đã chuẩn hóa).

Giai đoạn 3: Ánh xạ và sinh truy vấn cypher

Các cụm từ sau khi tách được ánh xạ ngược lại tên chuẩn trong KG thông qua conceptMap. Sau đó, hệ thống tự động sinh câu lệnh Cypher theo mẫu:

```
MATCH (d:Concept) WHERE d.name IN $fullConceptNames
OPTIONAL MATCH (d)-[r]->(related:Concept)
RETURN d, r, related
```

Giai đoạn 4: Thực thi và định dạng kết quả

Câu lệnh được gửi đến Neo4j; kết quả trả về bao gồm thông tin khái niệm (định nghĩa, ví dụ) và các mối quan hệ liên quan sẽ được đóng gói dưới dạng JSON để hiển thị trực quan cho người dùng.

2.3.3. Ví dụ minh họa quy trình

Giả sử người dùng đặt câu hỏi: “SQL hỗ trợ những gì trong truy vấn dữ liệu?”. Bảng 1 mô tả quy trình hệ thống hoạt động và kết quả trả về cho người dùng được minh họa trong Hình 5.

Bảng 1. Minh họa hoạt động của quy trình tra cứu

Bước	Hoạt động	Kết quả trung gian
1	Chuẩn hóa văn bản	“sql hỗ trợ những gì trong truy vấn dữ liệu?”
2	So khớp Bi-MM	[“sql”, “truy vấn dữ liệu”]
3	Ánh xạ Concept	[“SQL”, “Truy vấn (Query)”]
4	Sinh truy vấn	MATCH (d:Concept) WHERE d.name IN [“SQL”, “Truy vấn (Query)”] OPTIONAL MATCH (d)-[r]->(related:Concept) RETURN d, r, related
5	Trả kết quả	JSON chứa định nghĩa SQL, truy vấn và các quan hệ (HỖ_TRỢ, SỬ_DỤNG,...) Dữ liệu được định dạng thành JSON và trả cho người dùng. { name: 'Truy vấn (Query)', definition: 'Yêu cầu lấy hoặc xử lý dữ liệu trong CSDL', example: “SELECT * FROM SINH_VIEN WHERE Lop = 'CNTT';”, relationships: [{ concept: 'Bảng (Table)', rel: 'TAC_DONG_LEN' }, { concept: 'Ngôn ngữ truy vấn có cấu trúc (SQL)', rel: 'SU_DUNG' }, { concept: 'SELECT', rel: 'CO_THE_LA' }, { concept: 'INSERT', rel: 'CO_THE_LA' }, { concept: 'UPDATE', rel: 'CO_THE_LA' }] }

sql hỗ trợ những gì trong truy vấn dữ liệu? Tìm kiếm Xóa

Mẹo: Nhập câu hoàn chỉnh chứa các khái niệm CSDL như: transaction, normalization, index, join, trigger, stored procedure...

Khái niệm đã nhận diện (2)

Ngôn ngữ truy vấn có cấu trúc (SQL) Truy vấn (Query)

Kết quả chi tiết (2 khái niệm)

Truy vấn (Query)

Định nghĩa:
Yêu cầu lấy hoặc xử lý dữ liệu trong CSDL.

Ví dụ:
SELECT * FROM SINH_VIEN WHERE Lop = 'CNTT';

Liên quan:

Ngôn ngữ truy vấn có cấu trúc (SQL)

Định nghĩa:
Ngôn ngữ dùng để tạo, truy vấn và thao tác dữ liệu trong CSDL.

Ví dụ:
Lệnh SELECT, INSERT, UPDATE, DELETE thuộc SQL.

Liên quan:

Hình 5. Minh họa tra cứu kiến thức môn học

3. Kết quả và bàn luận

3.1. Thiết kế thực nghiệm

Để đánh giá hiệu quả hệ thống, nhóm nghiên cứu đã xây dựng một bộ dữ liệu gồm: (i) đồ thị tri thức 70 khái niệm và 150 quan hệ ngữ nghĩa, đóng vai trò như một prototype được xây dựng thủ công để kiểm chứng kiến trúc trước khi mở rộng lên các đồ thị lớn hơn; (ii) tập 50 câu hỏi tiếng Việt được phân loại theo độ sâu suy luận (20 câu 1-hop, 20 câu 2-hop, 10 câu 3-hop), phân bố đều trên các chủ đề mô hình dữ liệu, chuẩn hóa, ràng buộc toàn vẹn, SQL, giao dịch và điều khiển đồng thời; (iii) bộ đáp án chuẩn do hai giảng viên độc lập xác nhận. Hệ thống được triển

khai trên máy chủ CPU tầm trung (8–16 GB RAM, Neo4j Desktop); mỗi câu hỏi được gửi 3 lần và lấy trung bình thời gian phản hồi để giảm sai số. Để so sánh nhóm thiết lập một đường cơ sở (baseline) là cơ chế tìm kiếm từ khóa trên tập PDF giáo trình và slide, trong đó người thử nghiệm dùng chính câu hỏi hoặc tập từ khóa tương ứng để tra cứu rồi đối chiếu kết quả với đáp án chuẩn.

3.2. Chỉ số đánh giá

Ba chỉ số chính được sử dụng trong quá trình đánh giá là:

• **Precision:**

$$P = \frac{TP}{TP + FP} \quad (2)$$

• **Recall:**

$$R = \frac{TP}{TP + FN} \quad (3)$$

• **F1-score:**

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (4)$$

Ngoài ra, bài báo còn sử dụng chỉ số Response time – thời gian phản hồi trung bình từ lúc người dùng gửi câu hỏi đến khi hệ thống nhận kết quả từ Neo4j và phản hồi lên giao diện – để đánh giá hiệu năng của hệ thống.

3.3. Kết quả định lượng

Bảng 2 tóm tắt kết quả so sánh giữa hai phương pháp. Kết quả đo đạc cho thấy hệ thống truy xuất ngữ nghĩa dựa trên KG đạt độ chính xác trung bình 92% trên 50 câu hỏi, trong khi phương pháp tra cứu từ khóa chỉ đạt khoảng 57%. Nguyên nhân: KG hiểu ngữ nghĩa relationships, không chỉ keyword matching.

Thời gian phản hồi trung bình trên KG xấp xỉ 150 ms và khá ổn định giữa các lần thử, nhờ sử dụng chỉ mục cùng các mẫu truy vấn Cypher tương đối đơn giản, qua đó đáp ứng tốt yêu cầu thời gian thực trong môi trường học tập tương tác.

Bảng 2. Kết quả so sánh giữa hai phương pháp

Phương pháp	Precision (%)	Recall (%)	F1-score (%)	Thời gian phản hồi TB (ms)
Tìm kiếm từ khóa (baseline)	64	51	57	Không ổn định (phụ thuộc người dùng)
Truy xuất dựa trên KG	94	90	92	150

Phân tích theo độ sâu suy luận (Bảng 3) cho thấy F1 lần lượt đạt 97% (1-hop), 91% (2-hop) và 81% (3-hop), phản ánh xu hướng suy giảm hiệu năng khi độ sâu tăng do Bi-MM khó nhận diện đầy đủ các thực thể; độ chính xác suy luận đa bước (gộp 2-hop và 3-hop) đạt xấp xỉ 88%.

Bảng 3. Hiệu năng hệ thống theo độ sâu suy luận

Loại câu hỏi	Số câu	F1-score (%)	Thời gian TB (ms)
1-hop (trực tiếp)	20	97	120
2-hop (suy luận 2 bước)	20	91	155
3-hop (suy luận 3 bước)	10	81	190

3.4. Phân tích và thảo luận

Khi phân loại theo dạng câu hỏi, có thể nhận thấy hệ thống hoạt động rất tốt với các câu hỏi định nghĩa và liệt kê thành phần/quan hệ trực tiếp giữa hai khái niệm, nhờ cấu trúc quan hệ trong KG được thiết kế rõ ràng. Tuy nhiên, trong 50 câu hỏi thử nghiệm, vẫn tồn tại 8% trường hợp (4 câu) hệ thống trả lời sai hoặc thiếu. Phân tích thủ công xác định ba nhóm nguyên nhân chính:

- Lỗi nhận diện thực thể (~50% lỗi): Do thuật toán Bi-MM phụ thuộc vào từ điển, các câu hỏi có biểu đạt tự nhiên phức tạp khiến hệ thống không nhận diện đủ thực thể và phải chuyển sang chế độ keyword fallback.
- Hạn chế về dữ liệu và suy luận (~25% lỗi): Một số câu hỏi đòi hỏi suy diễn qua nhiều cạnh đồ thị nhưng KG hiện tại (xây dựng thủ công) chưa bao phủ hết các quan hệ ẩn hoặc khái niệm mở rộng ngoài giáo trình.
- Lỗi sinh truy vấn Cypher (~25% lỗi): Sự cứng nhắc của các mẫu (template) truy vấn đôi khi không khớp với cấu trúc cú pháp đặc biệt của câu hỏi, khiến ý định truy vấn trên Neo4j bị diễn giải sai.

Bảng 4. So sánh các công trình quốc tế

Công trình	Miền tri thức	Ngôn ngữ	F1-score	Multi-hop
CourseKG [2]	Khoa học máy tính	Tiếng Anh	89%	82%
EDUKG [3]	K12 (phổ thông)	Tiếng Trung	85%	N/A
Đề xuất (Ours)	Cơ sở dữ liệu	Tiếng Việt	92%	88%

Bảng 4 so sánh với các công trình quốc tế cho thấy hệ thống đề xuất đạt F1 92% và độ chính xác multi-hop 88% trên miền CSDL tiếng Việt, nổi bật ở khả năng truy vấn thời gian thực trên dataset tiếng Việt với kiến trúc có thể mở rộng, nhưng vẫn còn hạn chế vì ontology được trích xuất thủ công và chưa tích hợp LLM. Kết quả này phản ánh đặc trưng của các hệ thống truy vấn ngữ nghĩa dựa trên luật/mẫu: rất chính xác với các câu hỏi “trong miền mẫu” nhưng kém linh hoạt trước sự đa dạng của ngôn ngữ tự nhiên; tuy vậy, trong bối cảnh môn Cơ sở dữ liệu với tập mẫu câu hỏi không quá lớn, cách tiếp cận này vẫn tỏ ra khả thi và hiệu quả khi triển khai thực tế.

Về khả năng mở rộng, mặc dù KG hiện tại hoạt động ổn định quanh 150 ms, phân tích lý thuyết cho thấy Neo4j sử dụng chỉ mục B-tree trên các thuộc tính name và id, giúp truy vấn đơn khái niệm có độ phức tạp xấp xỉ $O(\log N)$ và theo tài liệu kỹ thuật có thể duy trì thời gian phản hồi dưới 500 ms ngay cả khi mở rộng lên quy mô từ 1.000 đến 10.000 nút. Do các thí nghiệm chỉ tiết ở quy mô lớn hơn vượt quá giới hạn dung lượng bài báo, việc tự động hóa xây dựng KG và tích hợp các mô hình ngôn ngữ lớn (LLM) như PhoBERT hay GPT-4 để cải thiện nhận diện thực thể và sinh Cypher linh hoạt hơn được xác định là hướng phát triển trọng tâm trong nghiên cứu tiếp theo.

4. Kết luận

Nghiên cứu đã xây dựng và triển khai thành công đồ thị tri thức (KG) trên nền tảng Neo4j hỗ trợ tra cứu môn Cơ sở dữ liệu, đạt F1-score 92% (cao hơn khoảng 35% so với tìm kiếm từ khóa) và độ chính xác khoảng 88% cho các truy vấn multi-hop, qua đó cho thấy tiềm năng ứng dụng trong các hệ thống hỗ trợ học tập tiếng Việt cho ngành CNTT. Trong tương lai, nhóm sẽ (1) tích hợp các mô hình ngôn ngữ lớn để cải thiện xử lý ngôn ngữ tự nhiên, (2) tự động hóa làm giàu dữ liệu bằng trích xuất thông tin và cập nhật KG liên tục, (3) mở rộng KG sang nhiều môn học kỹ thuật với cơ chế chia sẻ tri thức, và (4) triển khai trên các nền tảng LMS nhằm hỗ trợ lộ trình học tập cá nhân hóa và đánh giá thích ứng cho sinh viên.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Nguồn ngân sách khoa học và công nghệ Trường Đại học Sư phạm Thành phố Hồ Chí Minh trong đề tài mã số CS.2025.19.27.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] X. Zou, H. Lin, J. Wu, C. Zheng, and Q. Guan, “Constructing a knowledge graph for the database course group via deep learning,” in *Proc. 2024 IEEE 8th Conf. Energy Internet Energy Syst. Integration (EI²)*, Changsha, China, Dec. 2024, pp. 1234-1240, doi: 10.1109/ICEIT61397.2024.10540737.

-
- [2] Y. Li, Y. Liang, R. Yang, J. Qiu, C. Zhang, and X. Zhang, "CourseKG: An educational knowledge graph based on course information for precision teaching," *Appl. Sci.*, vol. 14, no. 7, Apr. 2024, doi: 10.3390/app14072710.
- [3] B. Zhao *et al.*, "EDUKG: A heterogeneous sustainable K-12 educational knowledge graph," *arXiv preprint arXiv:2210.12228*, Oct. 2022. [Online]. Available: <https://arxiv.org/abs/2210.12228>. [Accessed Mar. 9, 2026].
- [4] K. Qu *et al.*, "A survey of knowledge graph approaches and applications in education," *Electronics*, vol. 13, no. 13, Jul. 2024, doi: 10.3390/electronics13132537.
- [5] A. Hogan *et al.*, "Knowledge Graphs," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1-37, May 2021, doi: 10.1145/3447772.
- [6] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić, "Introducing Wikidata to the Linked Data Web," in *Proc. 13th Int. Semantic Web Conf. (ISWC)*, Riva del Garda, Italy, 2014, pp. 50-65, doi: 10.1007/978-3-319-11964-9_4.
- [7] V. H. Nguyen *et al.*, "Integrating knowledge based on ontology and knowledge graph for chatbot in education query," *HCMUE J. Sci.*, vol. 21, no. 5, pp. 771-785, May 2024, doi: 10.54607/hcmue.js.21.5.4129(2024).
- [8] K. T. Tran and M. H. Dinh, "Leveraging large-scale language models to efficiently transform natural language into Cypher queries," (In Vietnamese), *HUFLIT Journal of Science*, vol. 9, no. 1, 2025, doi: 10.71091/2354-113X/248.
- [9] S. T. Sukumar, C.-H. Lung, M. Zaman, and R. Panday, "Knowledge Graph Generation and Application for Unstructured Data Using Data Processing Pipeline," *IEEE Access*, vol. 12, pp. 136759-136770, Jan. 2024, doi: 10.1109/access.2024.3462635.
- [10] D. Diefenbach, V. López, K. Singh, and P. Maret, "Core Techniques of Question Answering Systems over Knowledge Bases: A Survey," *Knowl. Inf. Syst.*, vol. 55, no. 3, pp. 529-569, Jun. 2018, doi: 10.1007/s10115-017-1100-y.
- [11] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu, "BioWordVec: Improving Biomedical Word Embeddings with Subword Information and MeSH," *Sci. Data*, vol. 6, May 2019, doi: 10.1038/s41597-019-0055-0.
- [12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguist. (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171-4186, doi: 10.18653/v1/N19-1423.