

NGHIÊN CỨU PHÁT TRIỂN CÔNG CỤ TỰ ĐỘNG CHUYỂN ĐỔI CƠ SỞ DỮ LIỆU QUAN HỆ SANG CƠ SỞ DỮ LIỆU PHI QUAN HỆ

RESEARCH AND DEVELOPMENT OF AUTOMATIC TOOLS TO CONVERT RELATIONAL DATABASE TO NON RELATIONAL DATABASE

Lê Đức Anh¹, Lê Sỹ Toàn¹,
Phạm Văn Hà^{2,*}

TÓM TẮT

Bài báo đưa ra giải pháp và công cụ chuyển đổi tự động từ CSDL quan hệ phổ biến hàng đầu là SQL Server sang CSDL phi quan hệ cũng phổ biến hàng đầu là MongoDB. Việc ánh xạ các bảng hàng, cột... được thể hiện rất rõ, trong đó có thể dễ dàng so sánh kết quả thu được từ bảng quy tắc ánh xạ lược đồ RDB sang NoSQL.

Từ khóa: Chuyển đổi, cơ sở dữ liệu quan hệ, cơ sở dữ liệu phi quan hệ.

ABSTRACT

This article offers solutions and tools for automatic conversion from the leading popular relationship database SQL Server to non-relational databases that are also the most popular mongoDB. The mapping of rows, columns... it is very clearly shown, in which it is easy to compare the results obtained from the RDB schema mapping rules table to NoSQL.

Keywords: Convert, relational database, non relational database.

¹Lớp ĐH Hệ thống thông tin 02- K13, Khoa CNTT, Trường Đại học Công nghiệp Hà Nội

²Khoa CNTT, Trường Đại học Công nghiệp Hà Nội

*Email: phamvanha@gmail.com

1. GIỚI THIỆU

Chuyển đổi cơ sở dữ liệu quan hệ sang cơ sở dữ liệu NoSQL không phải là một nhiệm vụ dễ dàng vì chưa có phương pháp di chuyển chính xác. Sau khi chuyển đổi, không có đánh giá về hiệu suất và khả năng của dữ liệu trong mô hình dữ liệu mới so với dữ liệu trong mô hình cũ.

Một số nhà nghiên cứu đã đề xuất các giải pháp chuyển đổi giữa RDBMS và NoSQL, đưa ra ưu nhược điểm của các phương pháp. Trong [1] đã thảo luận về một vài phương pháp di chuyển và đảo ngược từ RDBMS đến MongoDB.

Trong [2] đã thống kê và trình bày khá chi tiết về kỹ thuật sử dụng, thí nghiệm, kết quả và ưu điểm 7 cách tiếp cận các phương pháp di chuyển dữ liệu.

Trong [5], các tác giả đã lập sơ đồ liên kết một-một, một-nhiều và nhiều-nhiều các mối quan hệ trong RDB theo hai bước. Trong bước đầu tiên, khóa hàng cho họ cột được chọn dựa trên mô hình nhập dự kiến của người dùng. Sau đó, trong bước thứ hai, bảng có mối quan hệ với một bảng khác sẽ được hợp nhất thành một họ siêu cột.

Các tác giả đã sử dụng bốn bước để ánh xạ các mối quan hệ một-một và một-nhiều từ RDB thành HBase. Ban đầu, dữ liệu được chuyển đổi sang mức không chuẩn hóa, tiếp theo là hợp nhất các bảng liên kết, sau đó một khóa hàng được xác định là tối ưu cho các mẫu truy cập khác nhau, và cuối cùng duy trì việc lập chỉ mục trên các bảng HBase.

Cũng có một số công trình đề xuất chuyển đổi lược đồ từ RDB sang NoSQL dựa trên tài liệu. Các tác giả đã đề xuất một khuôn khổ để thực hiện một thuật toán đã sử dụng siêu dữ liệu được lưu trữ trong RDB để tự động chuyển đổi các thực thể và liên kết các mối quan hệ. Các tác giả đã sử dụng một ứng dụng độc lập có tên MigDB, để phân tích các bảng trong RDB, tạo tệp JSON dựa trên các bảng, sau đó chuyển tệp JSON tới mạng nơ-ron. Hơn nữa, mạng đã đưa ra quyết định về cấu trúc phù hợp nhất để ánh xạ tệp JSON, cho dù nó sẽ là một cấu trúc nhúng hay tham chiếu. Công việc này được thực hiện cho hiệp hội lập bản đồ chỉ các mối quan hệ.

Về việc chuyển đổi sang cơ sở dữ liệu NoSQL dựa trên đồ thị, các tác giả trong [5] ánh xạ mối quan hệ kết hợp một-nhiều từ RDB đến NoSQL dựa trên đồ thị bằng cách tạo hai nút, với khóa chính của một bên được chèn vào nhiều bên như một tham chiếu. Để lập bản đồ mối quan hệ nhiều-nhiều, bảng tham gia được tạo trong RDB bị xóa và các khóa chính trong bảng tham gia được chèn vào mỗi nút tham gia.

Trong [3], các tác giả đã trình bày việc chuyển đổi RDB sang một số họ NoSQL cụ thể là khóa-giá trị, cột, tài liệu và đồ thị. Các tác giả đã xác định các khái niệm của mỗi cơ sở dữ liệu bằng cách sử dụng các bộ giá trị xác định, sau đó sử dụng thuật toán để trình bày quá trình chuyển đổi.

Trong [6], các tác giả đã trình bày một bộ điều hợp dữ liệu được sử dụng để truy vấn và ánh xạ giữa cơ sở dữ liệu SQL và NoSQL. Bộ điều hợp cho phép truy vấn từ ứng dụng và xử lý việc chuyển đổi cơ sở dữ liệu tại cùng thời gian.

Trong báo cáo này, các tác giả trình bày về việc thực hiện chuyển đổi cơ sở dữ liệu quan hệ sang cơ sở dữ liệu phi quan hệ bằng các cách ánh xạ các bảng, hàng, cột, ràng buộc,... một cách rất rõ ràng.

2. QUY TẮC ÁNH XẠ GIỮA CƠ SỞ DỮ LIỆU SQL SERVER VÀ CƠ SỞ DỮ LIỆU MONGODB

Trong quá trình nghiên cứu chuyển đổi ánh xạ một số nhà nghiên cứu đã đưa ra các bộ quy tắc tuân theo quy trình chuyển đổi truyền thống trong RDB, bắt đầu từ mối quan hệ kết hợp với các ràng buộc theo sau là mối quan hệ kế thừa.

Gồm có 6 quy tắc ánh xạ:

- Chuyển đổi mỗi quan hệ kết hợp 1-1
- Chuyển đổi mỗi quan hệ kết hợp 1-n
- Chuyển đổi mỗi quan hệ kết hợp n-n
- Sự chuyên môn hóa trong mỗi quan hệ thừa kế
- Phép hợp trong mỗi quan hệ thừa kế
- Mỗi quan hệ kết hợp

3. CHUYỂN ĐỔI LƯỢC ĐỒ TỪ CƠ SỞ DỮ LIỆU QUAN HỆ SANG CSDL MONGODB

Bảng 1. Tóm tắt các quy tắc để ánh xạ lược đồ từ cơ sở dữ liệu RDB sang NoSQL

RDB Relationships	Column-Based			Document-Based		Graph-Based		
	Column	Column Family	Super Column Family	Embedding	Referencing	Node Label	Node Property	Relationship Property
One-to-one		Y		Y	Y	Y	Y	
One-to-many			Y	Y	Y	Y	Y	
Many-to-many		Y	Y	Y	Y	Y	Y	Y
Specialization		Y			Y	Y	Y	
Union	Y	Y		Y		Y	Y	
Aggregation			Y	Y	Y	Y	Y	

Quy trình chuyển đổi lược đồ từ RDBMs sang MongoDB gồm các bước sau:

Bước 1: Xác định mục tiêu

Bước 2: Chuyển đổi dữ liệu

Bước 3: Kiểm tra và đánh giá

Bước 1: Xác định mục tiêu

Dựa vào lược đồ RDBMs có sẵn ta thực hiện các công việc sau:

- Xác định các thực thể dữ liệu
- Xác định các Index (nếu có)
- Phân loại các nhóm thực thể chính

Khi xác định mục tiêu cần quan tâm đến các vấn đề sau đây:

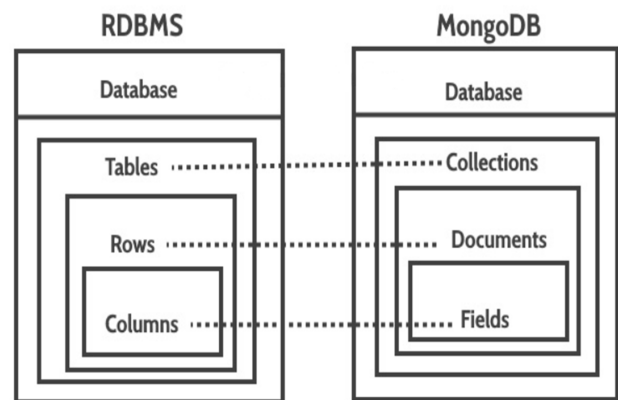
- Đối tượng sử dụng
- Phạm vi sử dụng
- Môi trường hoạt động

- Hệ thống người dùng
- Giao diện quản lý ứng dụng
- Chi phí chuyển đổi

Bước 2: Chuyển đổi dữ liệu

Với bất kỳ mô hình dữ liệu nào, mỗi tình huống chuyển đổi sẽ có sự khác nhau, tuy nhiên cũng có vài khái niệm chung có thể áp dụng cho hầu hết các sự chuyển đổi. Hình 1 dưới đây sẽ mô tả ánh xạ giữa cơ sở dữ liệu quan hệ và MongoDB.

- Các Collections (Bộ sưu tập) trong MongoDB sẽ tương ứng với các Tables (Bảng) trong RDBMS
- Các Document (Tài liệu) trong MongoDB tương đương với các Row (Hàng) trong RDBMS
- Các Fields (Trường) trong MongoDB tương đương với các Column (Cột) trong RDBMS
- Các Trường (cặp khóa-giá trị) được lưu trữ trong tài liệu, tài liệu được lưu trữ trong bộ sưu tập và bộ sưu tập được lưu trữ trong cơ sở dữ liệu.



Hình 1. Ánh xạ cơ sở dữ liệu quan hệ sang MongoDB

Document trong MongoDB nó tương tự như hàng trong RDBMs tuy nhiên sự khác biệt duy nhất là chúng ở định dạng JSON.

```
{
  name: "Chaitanya", ← Field: Value
  age: 30, ← Field: Value
  website: "beginnersbook.com", ← Field: Value
  hobbies: ["teaching", "watching tv"] ← Field: Value
}
```

Hình 2. Giao diện một tài liệu trong MongoDB

Dựa trên các mục tiêu chuyển đổi đã xác định trước mà người dùng lựa chọn một trong hai giải pháp sau:

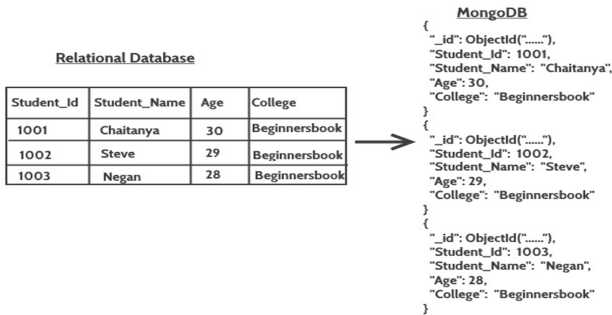
- Lựa chọn công cụ chuyển đổi có sẵn như Extract Transform Load
- Tự viết chương trình chuyển đổi dựa trên một số ánh xạ từ cấu trúc CSDL quan hệ sang cấu trúc MongoDB được trình bày.

Trong nghiên cứu này, tác giả lựa chọn nghiên cứu và xây dựng một công cụ chuyển đổi từ SQL Server sang MongoDB.

Ảnh xạ bảng, hàng, cột

Mỗi cơ sở dữ liệu trong MongoDB bao gồm các bộ sưu tập tương đương với cơ sở dữ liệu RDBMS bao gồm các bảng SQL.

Để rõ hơn quan sát hình 3 để thấy một bảng trong cơ sở dữ liệu quan hệ trông như thế nào trong MongoDB. Mỗi bộ sưu tập trong MongoDB lưu trữ dữ liệu dưới dạng tài liệu tương đương với các bảng trong SQL lưu trữ dữ liệu theo hàng. Trong khi một hàng lưu trữ dữ liệu trong tập hợp các cột của nó, thì một tài liệu có cấu trúc giống JSON (được gọi là BSON trong MongoDB).



Hình 3. Ví dụ ảnh xạ bảng trong RDBMS sang MongoDB

Như chúng ta thấy, các cột được biểu thị dưới dạng cặp khóa - giá trị (Định dạng JSON), các hàng được biểu thị dưới dạng tài liệu. MongoDB tự động chèn một trường `_id` (trường 12 byte) duy nhất vào mọi tài liệu, trường này đóng vai trò là khóa chính cho mỗi tài liệu.

Một điều thú vị khác về MongoDB là nó hỗ trợ lược đồ động, có nghĩa là một tài liệu của bộ sưu tập có thể có 4 trường trong khi tài liệu khác chỉ có 3 trường. Điều này không thể thực hiện được trong cơ sở dữ liệu quan hệ.

Lược đồ động (Dynamic Schema)

Các tài liệu khác nhau trong một bộ sưu tập có thể có các lược đồ khác nhau. Vì vậy, có thể trong MongoDB cho một tài liệu có năm trường và tài liệu kia có bảy trường. Các trường có thể dễ dàng thêm, bớt và sửa đổi bất cứ lúc nào. Ngoài ra, không có ràng buộc về kiểu dữ liệu của các trường. Do đó, có trường hợp một trường có thể chứa nguyên kiểu dữ liệu và ở trường hợp khác, nó có thể giữ một mảng.

Những khái niệm này phải có vẻ rất khác đối với người sử dụng đến từ nền tảng RDBMS, nơi cấu trúc bảng, cột, kiểu dữ liệu và quan hệ của chúng được xác định trước. Chức năng sử dụng lược đồ động này cho phép người quản trị tạo các tài liệu động tại thời điểm chạy.

Ví dụ: Hãy xem xét hai tài liệu sau trong cùng một bộ sưu tập nhưng có các lược đồ khác nhau (hình 4).

```

array (
  '_id' => new MongoId("5146bb52d8524270060001f2"),
  'address' => '123, Baker St, Dallas',
  'age' => new MongoInt32(31),
  'city' => 'Dallas',
  'dob' => '1990-01-03',
  'email' => 'richard@abc.com',
  'user_name' => 'Richard Peter',
)

array (
  '_id' => new MongoId("5146bb52d8524270060001f3"),
  'age' => new MongoInt32(25),
  'city' => 'Los Angeles',
  'email' => 'mark@abc.com',
  'gender' => 'Male',
  'occupation' => 'Doctor',
  'user_name' => 'Mark Hanks',
)
    
```

Hình 4. Ví dụ về lược đồ động

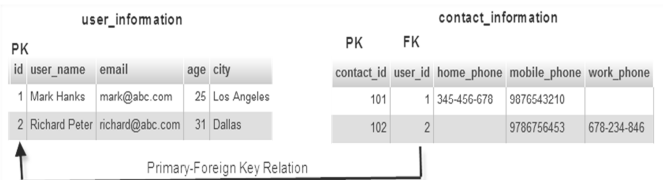
Tài liệu đầu tiên chứa các trường address và dob không có trong tài liệu thứ hai trong khi tài liệu thứ hai chứa các trường gender và occupation không có trong tài liệu đầu tiên. Nếu thiết kế điều này trong SQL thì sẽ giữ thêm bốn cột cho address, dob, gender và occupation, một số trong đó sẽ lưu trữ các giá trị rỗng (hoặc null), và do đó chiếm không gian không cần thiết.

Mô hình lược đồ động này là lý do tại sao cơ sở dữ liệu NoSQL có khả năng mở rộng cao về mặt thiết kế. Các lược đồ phức tạp khác nhau (phân cấp, cấu trúc cây,...) yêu cầu số lượng bảng RDBMS có thể được thiết kế hiệu quả bằng cách sử dụng các tài liệu như vậy. Một ví dụ điển hình sẽ là lưu trữ các bài đăng của người dùng, lượt thích, nhận xét của họ và các thông tin liên quan khác dưới dạng tài liệu. Nếu triển khai SQL cho cùng một lý tưởng sẽ có các bảng riêng biệt để lưu trữ các bài đăng, nhận xét và lượt thích trong khi tài liệu MongoDB có thể lưu trữ tất cả các thông tin này trong một tài liệu duy nhất.

Ảnh xạ phép nối và các mối quan hệ ràng buộc

Các mối quan hệ trong RDBMS đạt được bằng cách sử dụng các mối quan hệ khóa chính và khóa ngoài và truy vấn những người sử dụng các phép nối. Không có ảnh xạ đơn giản như vậy trong MongoDB nhưng các mối quan hệ ở đây được thiết kế bằng cách sử dụng các tài liệu nhúng và liên kết.

Hãy xem xét một ví dụ trong đó cần lưu trữ thông tin người dùng và thông tin liên hệ tương ứng. Một thiết kế SQL lý tưởng sẽ có hai bảng, chẳng hạn như `user_information` và `contact_information`, với các khóa chính `id` và `contact_id` như được hiển thị trong hình 5. Bảng `contact_information` cũng sẽ chứa một cột `user_id` sẽ là khóa ngoài liên kết đến trường `id` của bảng `user_information`.



Hình 5. Ví dụ hai bảng dữ liệu trong SQL

Bây giờ ta sẽ xem cách thiết kế các mối quan hệ như vậy trong MongoDB bằng cách sử dụng các phương pháp Liên kết tài liệu và Nhúng tài liệu. Hãy quan sát trong lược đồ SQL, ta thường thêm một cột (giống như `id` và `contact_id` trong trường hợp trên), cột này hoạt động như một cột chính cho bảng đó. Tuy nhiên, trong MongoDB thường sử dụng trường được tạo tự động `_id` làm khóa chính để xác định các tài liệu duy nhất.

Liên kết các tài liệu

Cách này sẽ sử dụng hai tập hợp `user_information` và `contact_information` cả hai đều có các trường `_id` duy nhất. Có một trường `user_id` trong tài liệu `contact_information` liên quan đến trường `_id` của tài liệu `user_information`. Lưu ý rằng trong MongoDB, các quan hệ và các hoạt động

tương ứng của chúng phải được thực hiện theo cách thủ công (ví dụ: thông qua mã) vì không có quy tắc và ràng buộc khóa ngoại nào được áp dụng.

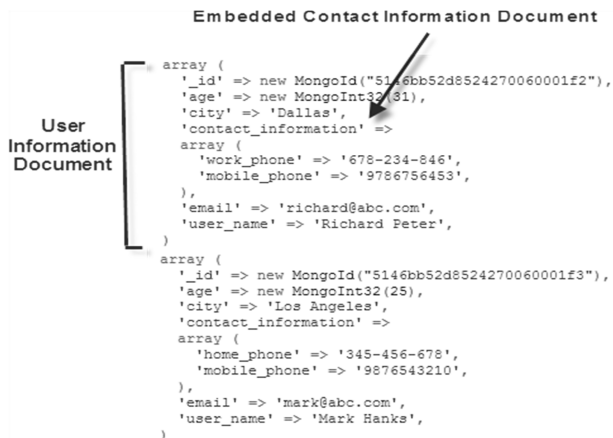


Hình 6. Mô tả liên kết các tài liệu

Các lĩnh vực `user_id` trong tài liệu chỉ đơn giản là một lĩnh vực chứa một số dữ liệu và tất cả các logic liên kết với nó phải được thực hiện. Ví dụ: ngay cả khi chèn một số `user_id` vào tài liệu `contact_information` không tồn tại trong bộ sưu tập `user_information`, MongoDB sẽ không gây ra bất kỳ lỗi nào nói rằng `user_id` không tìm thấy tương ứng trong bộ sưu tập `user_information` (không giống như SQL nơi đây sẽ là một ràng buộc khóa ngoại không hợp lệ).

Nhúng tài liệu

Cách tiếp cận thứ hai là nhúng tài liệu `contact_information` vào bên trong tài liệu `user_information` như hình dưới



Hình 7. Mô tả nhúng tài liệu

Trong ví dụ trên đã trình bày việc nhúng một tài liệu thông tin liên hệ vào bên trong thông tin người dùng. Theo cách tương tự, các tài liệu phức tạp lớn và dữ liệu phân cấp có thể được nhúng như vậy vào các thực thể có liên quan.

Ngoài ra, việc sử dụng cách tiếp cận nào trong số hai cách tiếp cận là Liên kết và Nhúng thì cần phụ thuộc vào từng tình huống cụ thể. Nếu dữ liệu được nhúng dự kiến sẽ tăng kích thước lớn hơn, tốt hơn nên sử dụng phương pháp Liên kết hơn là phương pháp Nhúng để tránh tài liệu trở nên quá lớn. Phương pháp nhúng thường được sử dụng trong các trường hợp phải nhúng một lượng thông tin hạn chế (như địa chỉ trong ví dụ trên).

Bước 3: Kiểm tra và đánh giá

Sau khi chuyển đổi dữ liệu cần kiểm tra, đánh giá một số nội dung sau:

- Cấu trúc dữ liệu;
- Thông tin dữ liệu;
- Các mối liên kết, ràng buộc;
- Tốc độ thực hiện;

4. KẾT LUẬN

Việc chuyển đổi từ cơ sở dữ liệu quan hệ sang cơ sở dữ liệu phi quan hệ mang ý nghĩa quan trọng trong bối cảnh các cơ sở dữ liệu quan hệ đã quá phổ biến và cần được kế thừa, trong khi cơ sở dữ liệu phi quan hệ lại ngày càng trở nên phổ biến.

Bài báo đã cho thấy sự thành công trong việc tạo ra bộ công cụ cho phép chuyển đổi từ cơ sở dữ liệu SQL Server sang MongoDB. Tuy nhiên còn một vấn đề mà đề tài nghiên cứu này gặp phải đó là việc tách và nhúng dữ liệu các khóa còn hạn chế.

Hướng phát triển tiếp theo của sẽ tiếp tục nghiên cứu và hoàn thiện, nâng cấp phiên bản cho công cụ để có thể tạo ra một công cụ hoàn chỉnh có thể nhúng dữ liệu, rút ngắn thời gian chuyển đổi, khắc phục tình trạng dư thừa dữ liệu khi chuyển đổi và phát triển công cụ để có thể chuyển đổi được các thành phần khác của cơ sở dữ liệu.

TÀI LIỆU THAM KHẢO

- [1]. Alae el Alami và Mohamed Bahaj, 2016. *Migration of a relational databases to NoSQL: The way forward*, 18-23.
- [2]. Y. Gu và các cộng sự, 2015. *Application of NoSQL database MongoDB*. 2015 IEEE International Conference on Consumer Electronics - Taiwan, tr. 158-159.
- [3]. T. Jia và các cộng sự, 2016. *Model Transformation and Data Migration from Relational Database to MongoDB*. 2016 IEEE International Congress on Big Data (BigData Congress), tr. 60-67.
- [4]. C. Lee và Y. Zheng, 2015. *Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases*. 2015 IEEE International Conference on Consumer Electronics - Taiwan, tr. 426-427.
- [5]. Ying-Ti Liao và các cộng sự, 2016. *Data adapter for querying and transformation between SQL and NoSQL database*. Future Generation Computer Systems. 65, tr. 111-121.
- [6]. Nguyễn Đình Thuận, Nguyễn Hữu Lộc, 2015. *Chuyển đổi lược đồ cơ sở dữ liệu SQL Server sang MongoDB*. Hội thảo quốc gia lần thứ XVIII Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông, TP Hồ Chí Minh.
- [7]. Nguyễn Văn Hòa, 2015. *Nghiên cứu về chuyển đổi lược đồ cơ sở dữ liệu quan hệ sang cơ sở dữ liệu NoSQL*. Đề tài luận văn cao học, ĐH Công nghệ TP Hồ Chí Minh.