

CHƯƠNG 5: MỘT SỐ MÔ HÌNH CSDL TIỀN TIẾN: CSDL PHI QUAN HỆ

Khoa Khoa học và kỹ thuật thông tin
Bộ môn Thiết bị di động và Công nghệ Web

Nội dung

1. Đặc điểm NoSQL.
2. Các mô hình NoSQL.
3. Chuyển từ mô hình SQL sang NoSQL.
4. Các CSDL NoSQL.

Vai trò của NoSQL

Đặt vấn đề

- Với sự phát triển Internet, dữ liệu xung quanh chúng ta được có được lớn hơn bao giờ hết.
- Mọi loại dữ liệu kiểu: chuỗi, số, âm thanh, hình ảnh có thể được đưa về dạng kỹ thuật số để bất kỳ máy tính nào cũng có thể lưu trữ, xử lý và chuyển tiếp cho nhiều người.
- Sự phát triển của mạng xã hội, cho phép người dùng tự do tạo các nội dung tương tác, làm tốc độ tăng trưởng khối lượng dữ liệu là cực lớn.
- Khối lượng dữ liệu tăng lên quá nhanh, vượt qua giới hạn xử lý của các hệ quản trị cơ sở dữ liệu truyền thống.

Vấn đề (tt)

- Việc lưu trữ và khai thác lượng dữ liệu khổng lồ này là một trong các thử thách lớn mà chúng ta gặp phải trong xã hội hiện đại.
- Các hệ cơ sở dữ liệu quan hệ hiện tại bộc lộ những hạn chế.
- Do đó, trong những năm gần đây, nhiều loại CSDL NoSQL được nghiên cứu.
- Những CSDL này đặc biệt thích hợp cho các ứng dụng cực lớn, giảm thiểu tối đa các phép tính toán, tác vụ đọc-ghi với khả năng chịu tải, chịu lỗi cao nhưng đòi hỏi tài nguyên phần cứng khá thấp.

ĐẶC ĐIỂM CỦA NOSQL

- NoSQL là một loại CSDL có các đặc tính sau:
 - + Không ràng buộc.
 - + Phân tán.
 - + Mã nguồn mở.
 - + Có khả năng mở rộng theo chiều ngang.
 - + Lược đồ tự do.
- NoSQL có thể lưu trữ xử lý dữ liệu từ một lượng rất nhỏ cho đến hàng petabytes, trong một hệ thống chịu tải, chịu lỗi cao và đáp ứng thời gian thực

SQL và NoSQL

Bảng 1. So sánh sự giống nhau và khác nhau giữa SQL và NoSQL

	SQL	NoSQL
Loại CSDL	Mỗi loại CSDL SQL có các biến thể nhỏ	Nhiều loại CSDL NoSQL khác nhau, bao gồm: CSDL cặp khóa – giá trị, CSDL hướng tài liệu, CSDL hướng cột và CSDL hướng đồ thị
Mục đích phát triển	Đáp ứng nhu cầu lưu trữ đầu tiên của các ứng dụng	Đáp ứng nhu cầu vượt qua ngoài khả năng đáp ứng của SQL, đặc biệt quan tâm đến quy mô của dữ liệu, phát triển và lưu trữ dữ liệu phi cấu trúc.
Các hệ CSDL điển hình	Microsoft SQL Server, MySQL, Postgre, Oracle Database	MongoDB, Cassandra, Hbase, Neo4j
Mô hình dữ liệu	Cấu trúc lưu trữ xây dựng trước khi lưu trữ dữ liệu	Không cần xây dựng sẵn cấu trúc lưu trữ, việc cần làm là tiến hành lưu trữ dữ liệu vào
Khả năng mở rộng	Chỉ có thể mở rộng theo chiều dọc	Có thể mở rộng theo chiều dọc, chiều ngang đồng thời hỗ trợ công nghệ điện toán đám mây
Mã nguồn	Cả đóng và mở	Chỉ có mã nguồn mở
Hỗ trợ	Hỗ trợ tốt cho khách hàng	Hỗ trợ trong một số trường với với một

ACID

- ACID là viết tắt của cụm từ
 - + Atomicity (nguyên tử).
 - + Consistency (nhất quán).
 - + Isolation (Cô lập).
 - + Durability (Lâu bền).
- Trong cơ sở dữ liệu NoSQL, các nguyên tắc của các mô hình ACID là quá mức cần thiết → trên thực tế nó đã cản trở hoạt động của các cơ sở dữ liệu.
- CSDL NoSQL thỏa mãn các đặc tính của ACID là vô cùng khó khăn → Consistency và Isolation bị thu hồi.

Nguyên tắc của NoSQL – Nguyên tắc BASE

— NoSQL dựa vào một mô hình nhẹ nhàng hơn, thích hợp hơn, và kết quả là chúng ta có phương pháp tiếp cận mới BASE gồm ba nguyên tắc:

+ **Basic Availability:** Tính sẵn có cơ bản.

- Một ứng dụng làm việc cơ bản tất cả thời gian (xuyên suốt).

+ **Soft State:** Trạng thái mềm, linh hoạt.

- Không cần phải nhất quán trên tất cả các thời gian hoạt động.

+ **Eventual Consistency:** Nhất quán cuối.

- Đạt được trạng thái cuối nhất quán.

➔ Là cách tiếp cận để quản lý dữ liệu phi cấu trúc.

Ưu điểm

- Đáp ứng được sự tăng trưởng của dữ liệu lớn.
- Truy xuất dữ liệu lớn với tốc độ cao.
- Dữ liệu đa dạng, có cấu trúc, bán cấu trúc hoặc phi cấu trúc.
- Dữ liệu phức tạp, được lưu trữ và quản lý tại các trung tâm lưu trữ khác.
- Cần ít tài nguyên và phần cứng của máy chủ.
- Hỗ trợ chỉ mục tất cả các thuộc tính.
- Mã nguồn mở.
- Có thể mở rộng theo chiều dọc.

Ưu điểm (tt)

- NoSQL được các hãng lớn sử dụng: Các công ty như Amazon, BBC, Facebook và Google dựa vào các CSDL NoSQL.
- NoSQL và đám mây (cloud technology) là một sự trùng khớp tự nhiên, chúng có khả năng tận dụng được việc cung cấp mềm dẻo của các dịch vụ lưu trữ trên đám mây (cloud storage).
- Các CSDL NoSQL hầu hết sử dụng bộ nhớ qua ổ đĩa như là vị trí ghi đầu tiên - vì thế ngăn ngừa được sự thực thi không ổn định của thao tác I/O.

Nhược điểm

- Sự tin tưởng chưa cao đối với nhiều doanh nghiệp.
 - + CSDL truyền thống vẫn là lựa chọn số một.
- Tính mới mẻ của NoSQL có nghĩa là không có nhiều lập trình viên và người quản trị mà biết công nghệ này.
- Những vấn đề về tính tương thích: Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng (API) riêng của mình, các giao diện truy vấn riêng biệt.
- Khó khăn trong việc lưu trữ các dữ liệu mang nội dung nghiệp vụ phức tạp.

Các mô hình NoSQL

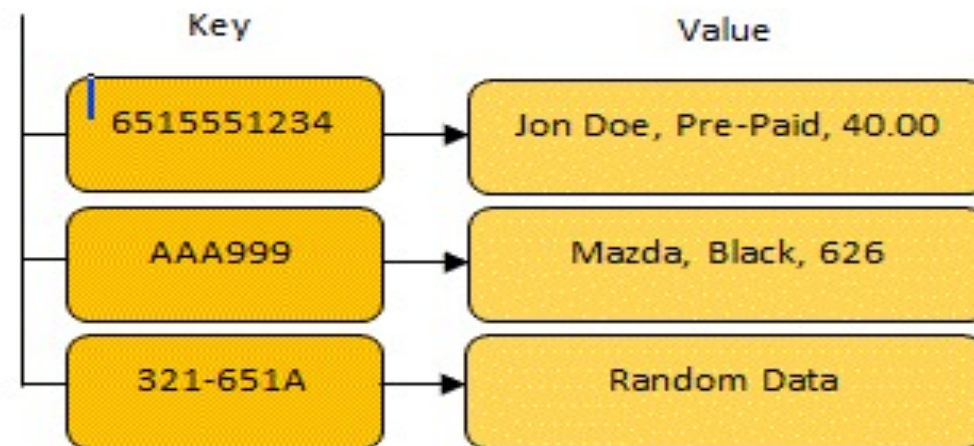
Các mô hình phi quan hệ

1. Hướng tài liệu (Document).
2. Khoá – giá trị (key – value).
3. Hướng cột (Column).
4. Đồ thị (Graph).

Khoá – giá trị (key-value)

- Dữ liệu được xác định bằng một khoá duy nhất (Key).
- Các giá trị (value) hoàn toàn tách biệt và không phụ thuộc vào nhau.
- Cấu trúc dữ liệu rất đơn giản nên cơ sở dữ liệu cặp khoá – giá trị hoàn toàn không có lược đồ.
- Giá trị mới có thể được thêm vào trong lúc hệ thống đang chạy mà không gây ra bất cứ xung đột dữ liệu nào.
- Cơ sở dữ liệu cặp khoá – giá trị hữu ích cho các xử lý đơn giản, chỉ phụ thuộc vào thuộc tính khoá.
- Ứng dụng: từ điển, truy xuất bộ đệm.

Khoá – giá trị (key-value)



Hướng tài liệu (Document)

- Mô hình Hướng tài liệu được thiết kế dùng để lưu trữ, truy xuất và quản lý dữ liệu có dạng tài liệu hay dữ liệu bán cấu trúc hoặc thông tin.
- Khái niệm về quan hệ (relations hay bảng) trong những hệ thống này được thiết kế xung quanh một khái niệm trừu tượng gọi là tài liệu (document).
- Sử dụng **key-document** (hoặc **key-value**) để lấy một tài liệu, cơ sở dữ liệu sẽ cung cấp một API hoặc ngôn ngữ truy vấn cho phép bạn lấy các tài liệu dựa trên nội dung.
- Ứng dụng: Các hệ thống lưu trữ nội dung mạng xã hội, trang web.

Hướng tài liệu (Document)

Collection			
<table border="1"><thead><tr><th>Document</th></tr></thead><tbody><tr><td>Name: "Rick"</td></tr><tr><td>Age: 23</td></tr></tbody></table>	Document	Name: "Rick"	Age: 23
Document			
Name: "Rick"			
Age: 23			
<table border="1"><thead><tr><th>Document</th></tr></thead><tbody><tr><td>Item ID: 12</td></tr><tr><td>Amount: 45</td></tr></tbody></table>	Document	Item ID: 12	Amount: 45
Document			
Item ID: 12			
Amount: 45			

Hướng cột (Column)

- Mô hình hướng cột xem xét nhiều trường hợp khác nhau của các thuộc tính chỉ chứa một cặp giá trị cần thiết trong mỗi dòng, còn lại là các giá trị null.
- Nhìn chung cơ sở dữ liệu hướng cột có nhiều điểm tương đồng với cơ sở dữ liệu quan hệ, nếu nhìn từ bên ngoài, nhưng thật sự có nhiều khác biệt lớn từ bên trong.
 - + Điểm khác biệt chính là việc lưu trữ null đối với các thuộc tính không cần thiết.
- Ngoài ra, một trong những khác biệt đó chính khác là việc lưu trữ dữ liệu theo cột thay vì theo dòng như trong cơ sở dữ liệu quan hệ → số thuộc tính có thể không cần phải xác định trước.

Hướng cột (column)

ID	Column		
1	Tên	Website	
	Nam	www.vnexpress.com	<null>
2	Tên	Email	Website
	Mai	mai@gmail.com	Dantri.com.vn



2 cột

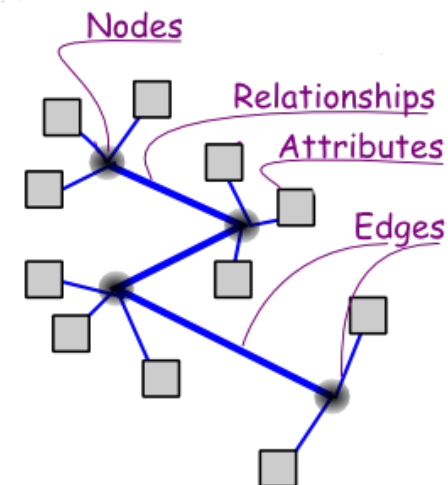
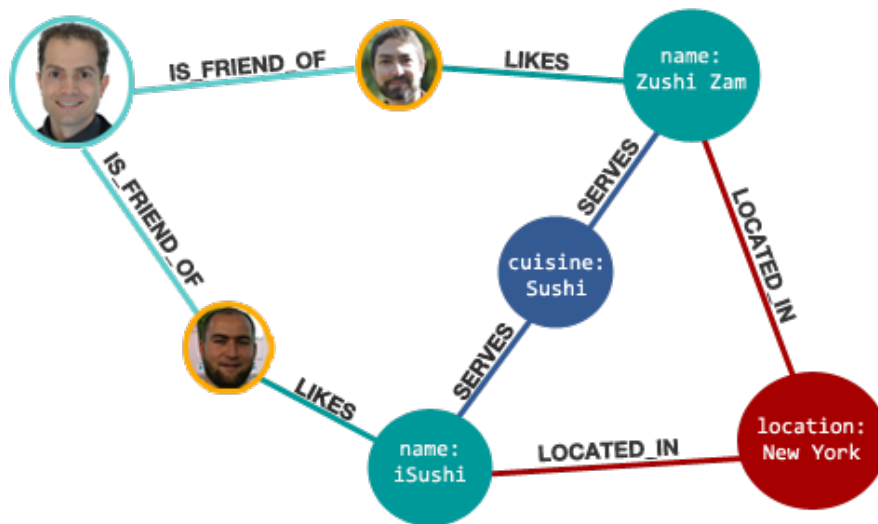


3 cột

Đồ thị (Graph)

- Mô hình đồ thị chuyên dùng trong quản lý dữ liệu với nhiều liên kết (quan hệ).
- Mỗi thực thể (instance) trong CSDL quan hệ sẽ ứng với một nút (node), và quan hệ giữa các thực thể sẽ được biểu diễn bởi các cạnh (edge).
- Nút (node) và cạnh (edge) bao gồm các đối tượng chứa các cặp khoá – giá trị. Tầm vực (scope) của các cặp khoá – giá trị được định nghĩa trong lược đồ, nên các ràng buộc phức tạp được mô tả dễ dàng.

Đồ thị (Graph)



Chuyển từ mô hình quan hệ sang NoSQL

Ví dụ: Mô hình quan hệ

Tacgia (#mstg,	tentg,	sdt,	email)	
TG01	Nguyen A	098731	a@gmail.com	
TG02	Nguyen B	098731	b@gmail.com	
Sach (#mssach,	tensach,	sotrang,	sotien,	msnxb)
S01	ABC	6	100000	DHQG
Nxb (#msnxb,	tennxb,	sdt-xb,	email-xb)	
NXB01	NXB-DHQG	0844643	nxb@gmail.com	
Tg-Sach(#mstg,	#mssach,	nam-xb)		
TG01	S01	2019		
TG02	S01	2020		

Chuyển sang key-value

Key	Value
-----	-------

TACGIA

```
{  "TG01": "NguyenA_098731_a@gmail.com",
    "TG02": "NguyenB_098731_b@gmail.com"
}
```

SACH

```
{  "S01": "ABC _Trang:6_100000_NXB: DHQG"
}
```

NXB

```
{  "NXB01": "NXB-DHQG_DT:0844643_nxb@gmail.com"
}
```

Key	Value
-----	-------

TG_SACH

```
{
    "TG01_S01": "2019",
    "TG02_S01": "2020"
}
```

Chuyển sang Document (1)



TACGIA

```
[{  
  "MATG": "TG01",  
  "HOTEN": "NguyenA",  
  "SDT": "098731",  
  "Email": "a@gmail.com",  
  "SACH": ("MASACH": "S01", ..., "NAM": "2019")  
},  
{  
  "MATG": "TG02",  
  "HOTEN": "NguyenB",  
  "SDT": "098731",  
  "Email": "b@gmail.com",  
  "SACH": ("MASACH": "S01", ..., "NAM": "2020")  
}]
```

Chuyển sang Document (2)

SACH

```
[  
{  
  "MASACH": "S01",  
  "TENSACH": "ABC",  
  "SOTRANG": 6,  
  "SOTIEN": "100000",  
  "NXB": ("MANXB": "NXB01", "TEN": "NXB-DHQG",  
          "DT": "0844643", "EMAIL": "nxb@gmail.com"),  
  "TACGIA": [{ ("MATG": "TG01", ...), "NAM": "2019"},  
              { ("MATG": "TG02", ...), "NAM": "2020"}]  
}  
]
```

Chuyển sang Document (3)

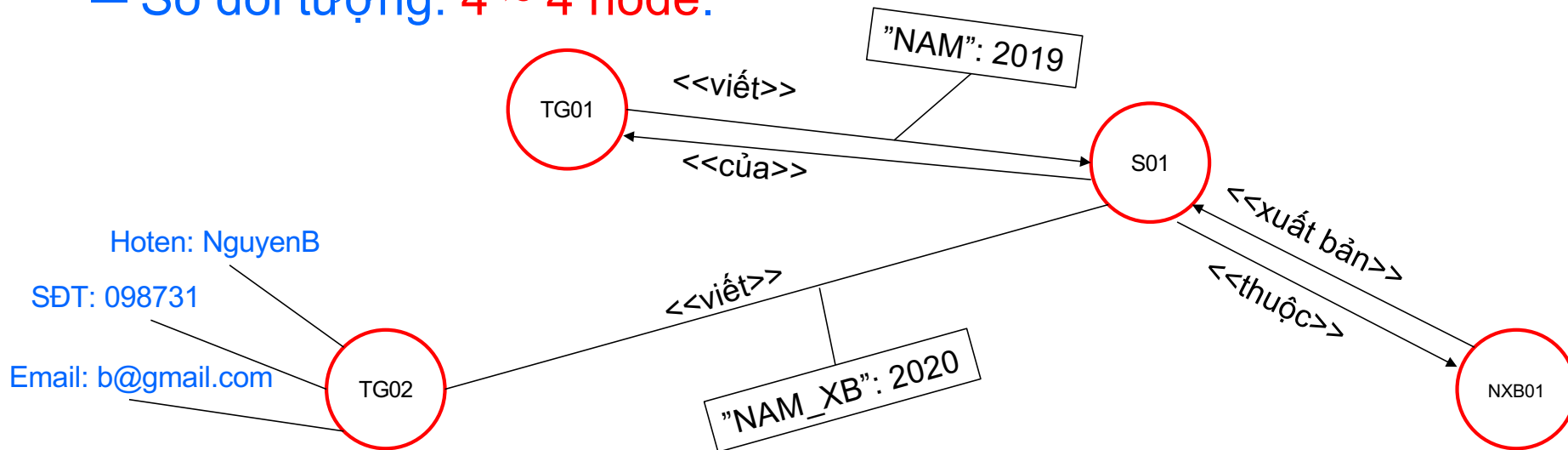
```
NXB
[
{
  "TEN": "NXB-DHQG",
  "MANXB": "NXB01",
  "DT": "0844643"
  "EMAIL": "nxb@gmail.com",
  "SACH": { ("MASACH": "S01",...) }
}
```

Chuyển sang Column

ID	Column			
Tacgia (#mstg,	tentg,	sdt,	email)	
	Ho Ten			
TG01	Nguyen A	098731	<u>a@gmail.com</u>	4 cột
	HoVaTen			
TG02	Nguyen B	098731	<u>b@gmail.com</u>	3 cột
Sach (#mssach,	tensach,	sotrang,	sotien,	msnxb)
S01	ABC	6	100000	DHQG
Nxb (#msnxb,	tennxb,	sdt-xb,	email-xb)	
NXB01	NXB-DHQG	0844643	<u>nxb@gmail.com</u>	
Tg-Sach (#mstg, #mssach,		nam-xb)		
TG01,S01		2019		
TG02,S01		2020		

Chuyển sang Graph

- Đối tượng: TACGIA, SACH, NXB.
- Số đối tượng: 4 ~ 4 node.



So sánh giữa các mô hình

Bảng 3. Sự giống và khác nhau giữa 4 loại CSDL NoSQL

Loại CSDL NoSQL		Khả năng truy vấn		Quản lý đồng thời		Phân vùng		Nhân bản	
		Java API	Ngôn ngữ truy vấn	Khóa	Khóa tích cực	Dây cơ sở	Bảng băm	Đọc	Ghi
Cặp khóa – giá trị	Redis	Có	Không	Không	Có	Không	Có	Không	Không
	Membase	Có	Không	Không	Có	Không	Có	Không	Không
Hướng tài liệu	MongoDB	Có	Không	Không	Không	Có	Không	Không	Không
	CouchDB	Có	Không	Không	Không	Không	Có	Có	Có
Hướng cột	Cassandra	Có	Có	Không	Không	Không	Có	Không	Không
	Hbase	Có	Có	Có	Không	Có	Không	Không	Không
Đồ thị	Neo4j	Có	Có	Có	Không	Không	Không	Có	Có
	Graph DB	Có	Không	Không	Không	Không	Không	Không	Không

<https://arxiv.org/ftp/arxiv/papers/1307/1307.0191.pdf>

CSDL phi quan hệ















Đặc điểm (1)

- **Mô hình dữ liệu phi quan hệ (Non-relational):** các mô hình này không có mối quan hệ ràng buộc lẫn nhau. Có thể có những cấu trúc dữ liệu phức tạp hơn, nhưng nó không cứng nhắc như mô hình dữ liệu quan hệ. Non-relational là khái niệm không sử dụng các ràng buộc dữ liệu cho nhất quán dữ liệu ở NoSQL database.
- **Lưu trữ phân tán (Distributed storage):** CSDL NoSQL được phân tán sang nhiều máy tính khác nhau, để cung cấp dữ liệu cho người dùng. Mỗi phần dữ liệu sau đó sẽ được nhân rộng trên một số lượng nhất định máy dự phòng với tính sẵn sàng đáp ứng cao.

Đặc điểm (2)

- **Nhất quán cuối (Eventual consistency):** Tính nhất quán của dữ liệu không cần phải đảm bảo ngay tức khắc sau mỗi tác vụ ghi. Một hệ thống phân tán chấp nhận những ảnh hưởng theo phương thức lan truyền và sau một khoảng thời gian (không phải ngay tức khắc), thay đổi sẽ đi đến mọi điểm trong hệ thống, tức là cuối cùng (eventually) dữ liệu trên hệ thống sẽ trở lại trạng thái nhất quán.
- **Khả năng mở rộng chiều dọc (Vertical scalable):** Khi dữ liệu lớn về lượng, phương pháp tăng cường khả năng lưu trữ và xử lý bằng việc cải tiến phần mềm và cải thiện phần cứng trên một máy tính đơn lẻ được gọi là khả năng mở rộng chiều dọc.

Các CSDL phi quan hệ thường gặp

Document Database	Graph Databases
  	 
Wide Column Stores	Key-Value Databases
    	   

Tham khảo: <https://db-engines.com/en/ranking>

TỔNG KẾT

1. CSDL phi quan hệ được thiết kế nhằm phá bỏ một số nguyên tắc nhất định của CSDL quan hệ → linh hoạt, uyển chuyển.
+ Consistency và Isolation bị thu hồi.
2. Các mô hình CSDL NoSQL thường gặp: Hướng tài liệu (document), hướng cột (column), khoá-giá trị (key-value) và đồ thị (graph).
3. Các CSDL phi quan hệ thường gặp: MongoDB (document), Cassandra (column), Redis (key-value), ...

TÀI LIỆU THAM KHẢO

1. Nguyễn Gia Tuấn Anh, Trương Châu Long, *Bài tập và bài giải SQL Server*, NXB Thanh niên (2005).
2. Đỗ Phúc, Nguyễn Đăng Ty, *Cơ sở dữ liệu*, NXB Đại học quốc gia TP HCM (2010).
3. Nguyễn Gia Tuấn Anh, Mai Văn Cường, Bùi Danh Hùng, *Cơ sở dữ liệu nâng cao*, NXB Đại học quốc gia TP HCM (2019).
4. Itzik Ben-Gan, *Microsoft SQL Server 2012- TSQL Fundamentals*.



Phụ lục: thực nghiệm so sánh CSDL quan hệ và phi quan hệ

So sánh giữa CSDL quan hệ và phi quan hệ

- Bài toán: So sánh thời gian thực thi của các thao tác truy xuất đọc/ghi vào CSDL.
- CSDL minh họa:
 - + CSDL quan hệ: **SQL**.
 - + CSDL phi quan hệ (hướng tài liệu): **MongoDB**.

Các đối tượng

Bảng 4. Các đối tượng thuộc ứng dụng minh họa

STT	Bộ sưu tập	Ý nghĩa
1	User	Lưu thông tin đăng nhập của user
2	UserInfo	Lưu thông tin chi tiết của user
3	Post	Lưu thông tin, nội dung bài viết
4	Comment	Lưu thông tin và nội dung của bình luận
5	Album	Lưu thông tin của album
6	Photo	Lưu thông tin và đường dẫn của hình ảnh
7	User Request	Lưu trữ thông tin của user yêu cầu kết bạn và user được kết bạn
8	Friends	Lưu trữ thông tin của các user là bạn của nhau
9	Like	Lưu trữ thông tin của hành động “Like”
10	TimeLine	Lưu trữ thông tin đăng trên “ dòng thời gian ”

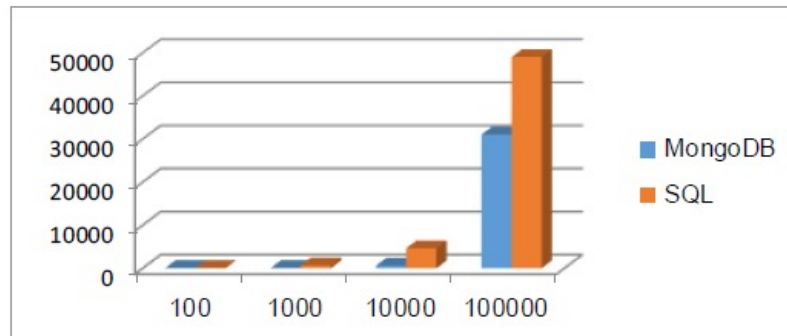


Cấu hình máy tính

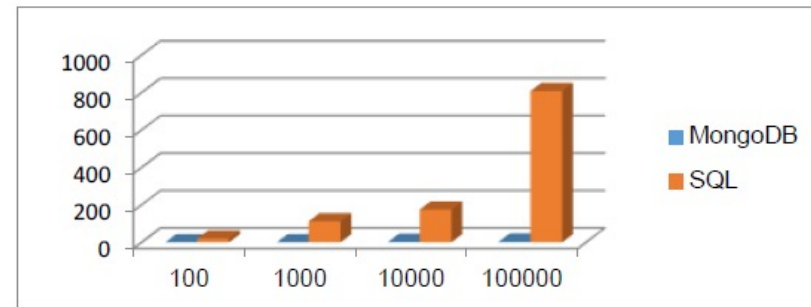
Bảng 5. Mô tả cấu hình máy tính thực nghiệm

Thành phần	Thông số
Hệ điều hành	Windows 8.1 Pro 64-bit
CPU	Intel® Core™ i5 – 2430M CPU @ 2.40GHz 2.40GHz
RAM	8.00GB
HDD	500GB 7200rpm sata
Hệ quản trị CSDL	MongoDB 3.0 và SQL Server 2014 express
GUI cho CSDL	Robotmongo 0.8.5 và SQL Server 2014 Management Studio

Kết quả (1)

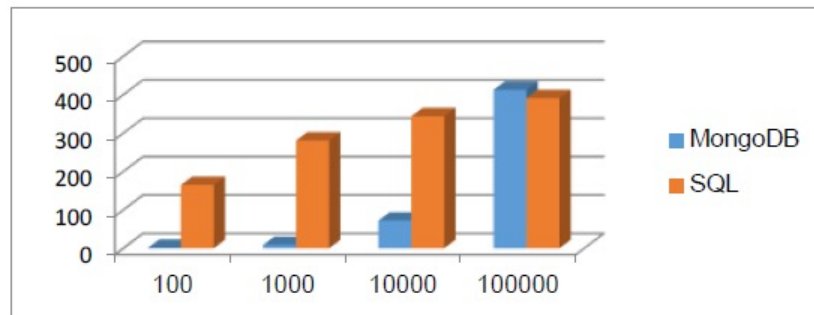


Hình 1. Biểu đồ chi phí thời gian đáp ứng khi thêm mới tài khoản

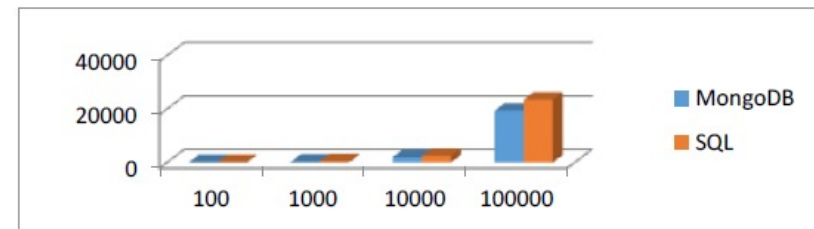


Hình 2. Biểu đồ chi phí thời gian đọc tài khoản người dùng

Kết quả (2)



Hình 3. Biểu đồ chi phí thời gian xóa tài khoản người dùng



Hình 4. Biểu đồ chi phí thời gian cập nhật tài khoản người dùng

Nhận xét

- Chi phí thời gian thực hiện xem của MongoDB là vượt xa so với MSSQL. Chi phí thời gian thực hiện thao tác thêm của MongoDB là vượt trội so với MSSQL. Chi phí thời gian thao tác cập nhật của MongoDB và MSSQL là khá cân bằng.
- Rõ ràng MogoDB rất phù hợp cho các ứng dụng có dữ liệu lớn, phục vụ cho 2 thao tác tìm kiếm và thêm. Đây cũng là đặc trưng của ứng dụng mạng xã hội.