

25 YEARS ANNIVERSARY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

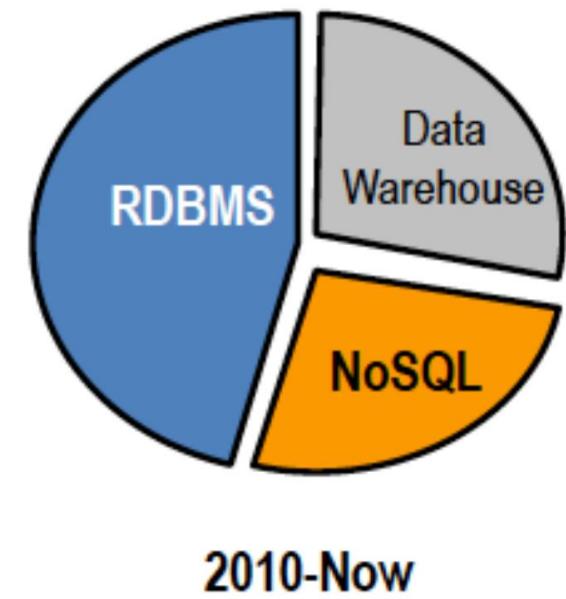
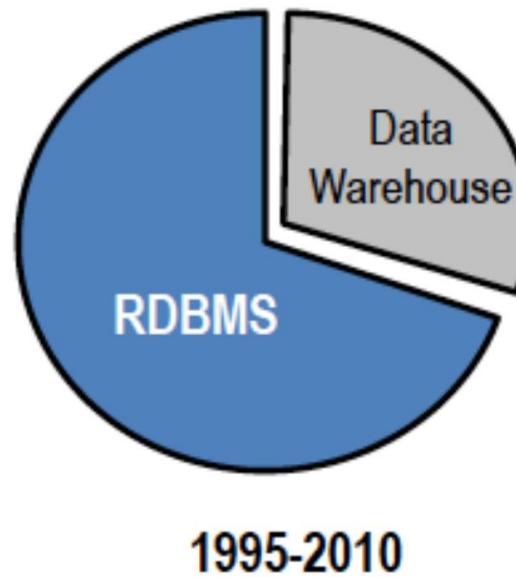
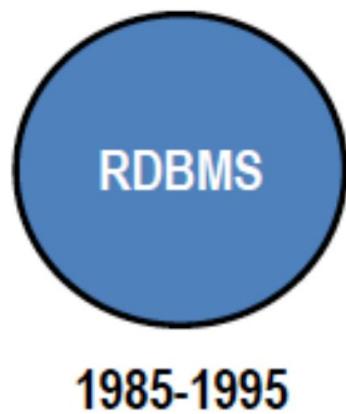


HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

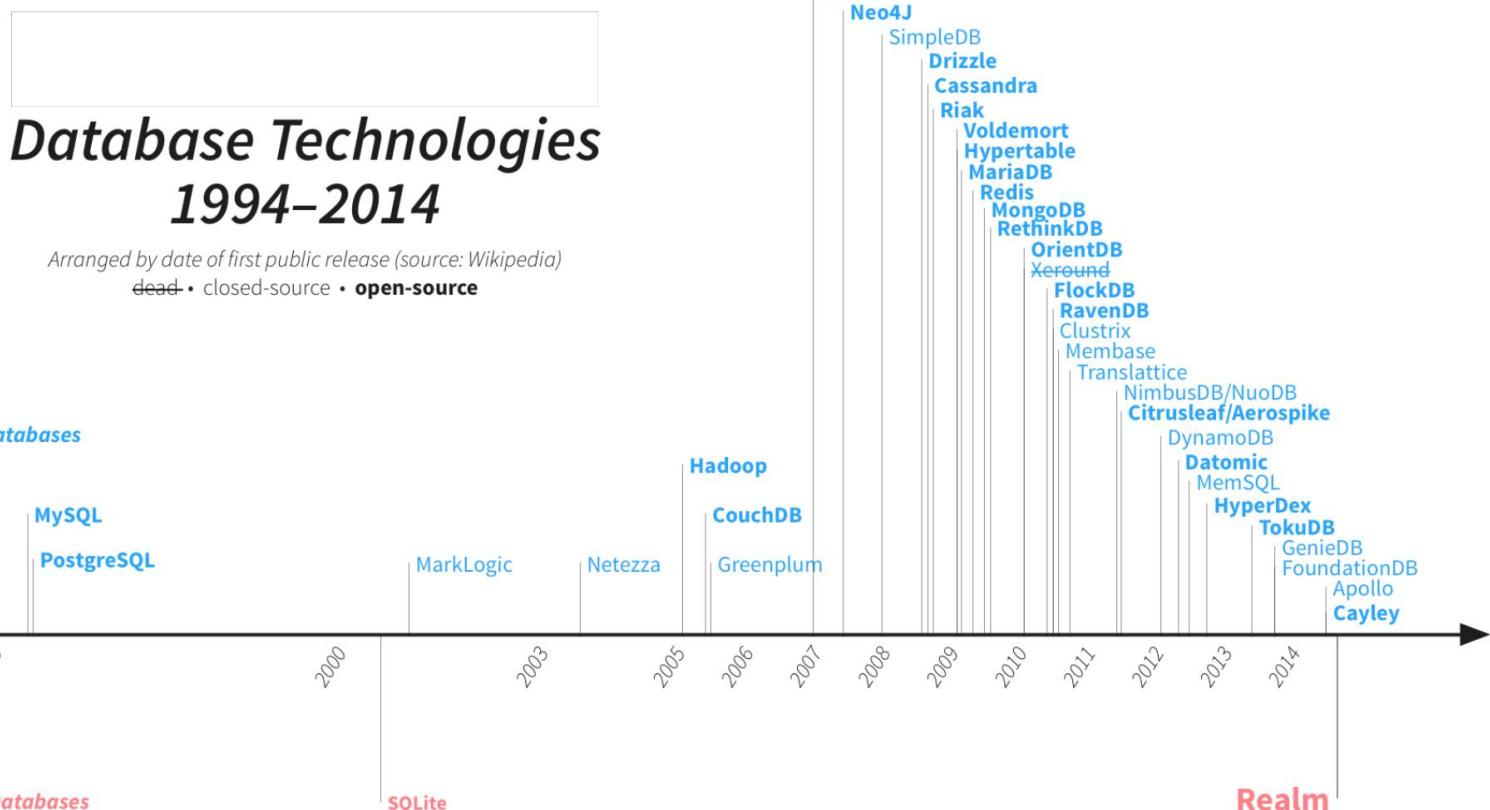
Chương 4

NoSQL - phần 1

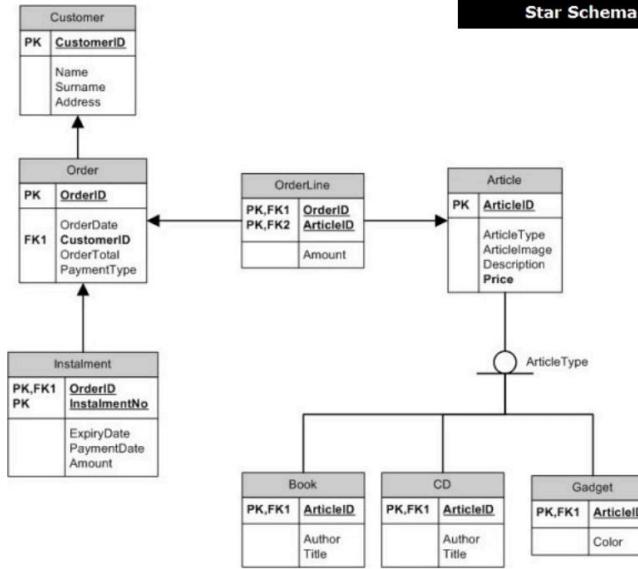
Kỷ nguyên của cơ sở dữ liệu



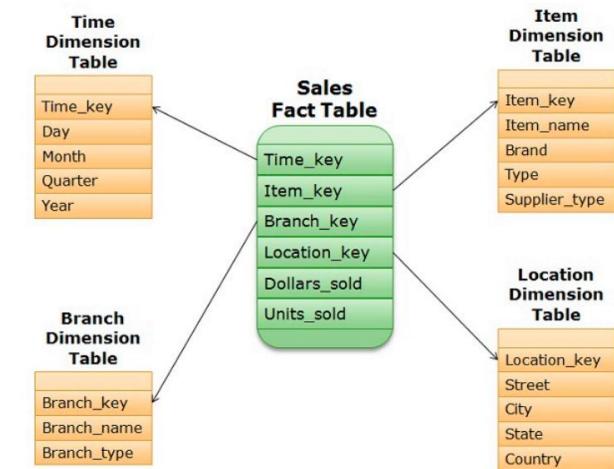
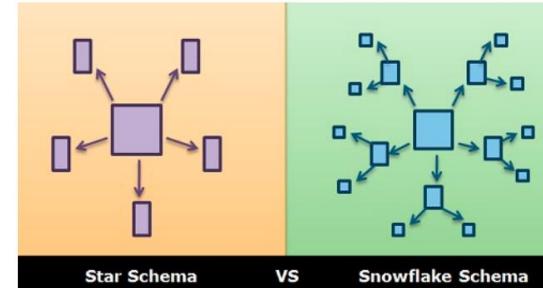
Kỷ nguyên của cơ sở dữ liệu



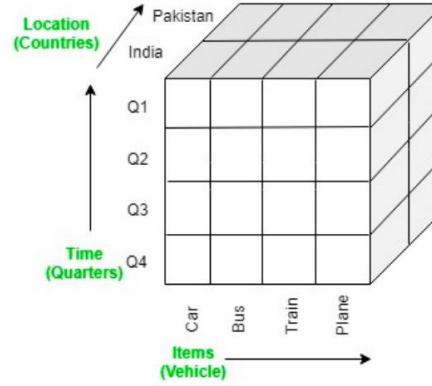
Trước NoSQL



OLTP

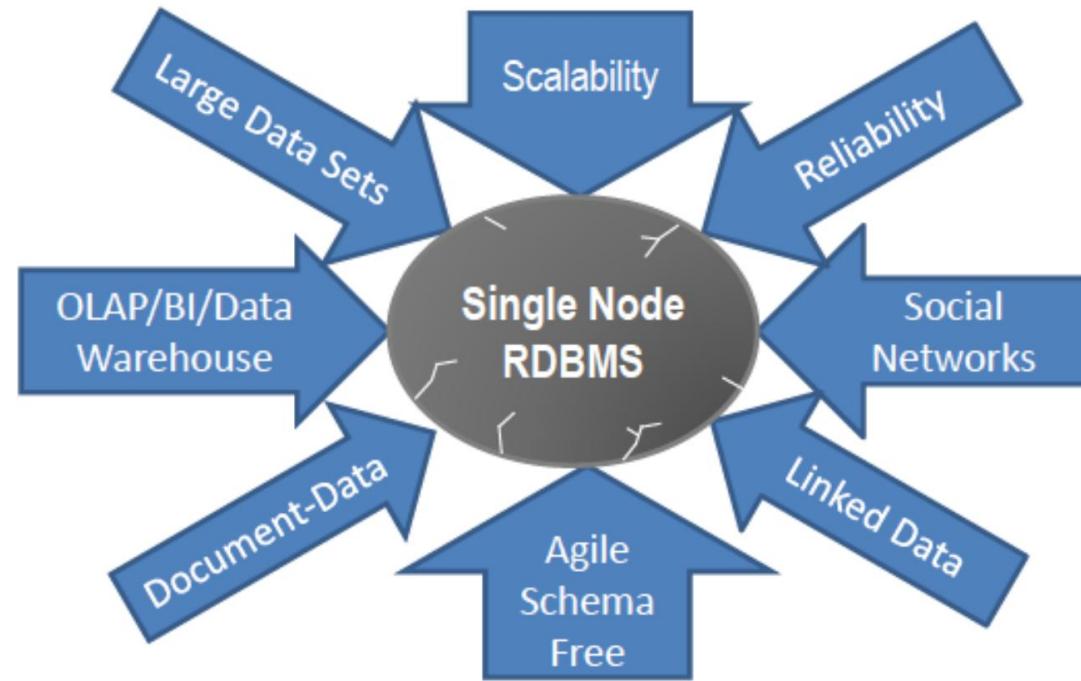


Sơ đồ ngôi sao



khối OLAP

RDBMS: một kích thước phù hợp với tất cả các nhu cầu



Hội nghị ICDE 2005

"One Size Fits All": An Idea Whose Time Has Come and Gone

Authors: [Michael Stonebraker](#) StreamBase Systems, Inc.
[Ugur Cetintemel](#) [Brown University and StreamBase Systems, Inc.](#)



2005 Article

Published in:

- Proceeding
 ICDE '05 Proceedings of the 21st International Conference on Data Engineering
 Pages 2-11

April 05 - 08, 2005

IEEE Computer Society Washington, DC, USA ©2005

[table of contents](#) ISBN:0-7695-2285-8 doi:>[10.1109/ICDE.2005.1](https://doi.org/10.1109/ICDE.2005.1)



Bibliometrics

- Citation Count: 73
- Downloads (cumulative): 0
- Downloads (12 Months): 0
- Downloads (6 Weeks): 0

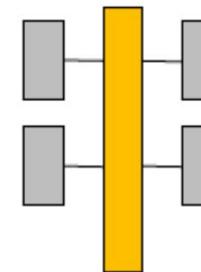
25 năm phát triển DBMS thương mại gần đây có thể được tóm tắt trong một cụm từ duy nhất: "một kích thước phù hợp với tất cả". Cụm từ này ám chỉ thực tế là **kiến trúc DBMS truyền thống** (ban đầu được thiết kế và tối ưu hóa cho xử lý dữ liệu doanh nghiệp) đã được sử dụng để hỗ trợ nhiều ứng dụng lấy dữ liệu làm trung tâm với các đặc điểm và yêu cầu rất khác nhau. Trong bài báo này, chúng tôi lập luận rằng khái niệm này không còn áp dụng được cho thị trường cơ sở dữ liệu nữa và thế giới thương mại sẽ phân chia thành một tập hợp các công cụ cơ sở dữ liệu độc lập ...

Sau NoSQL

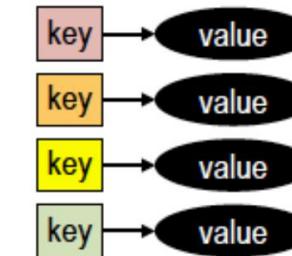
Relational



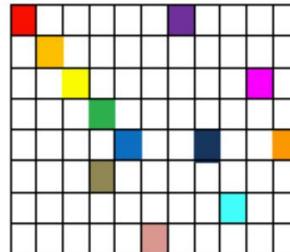
Analytical (OLAP)



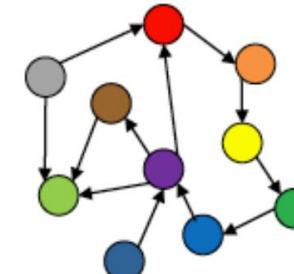
Key-Value



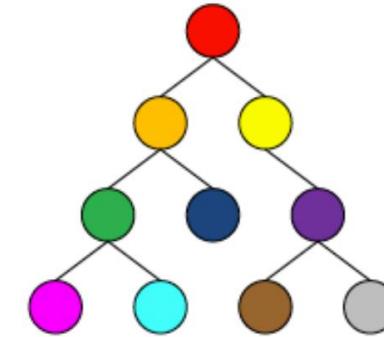
Column-Family



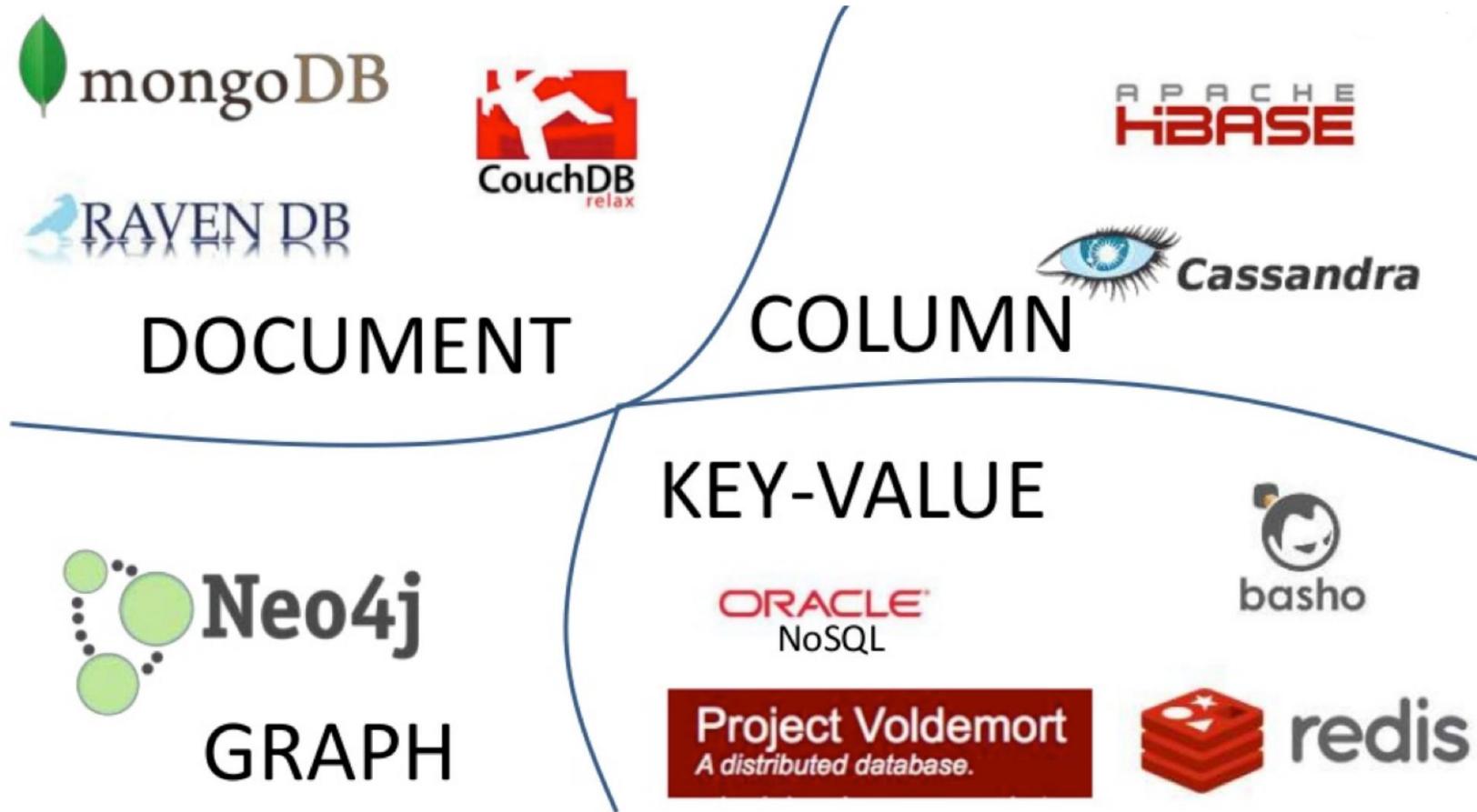
Graph



Document

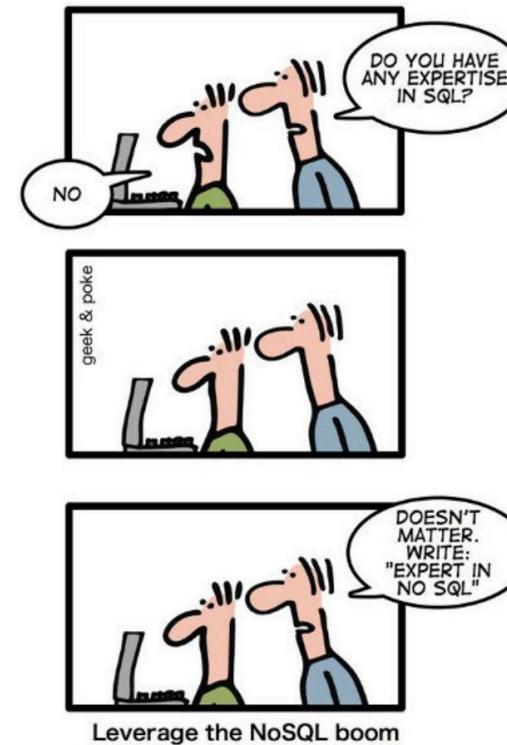


Cảnh quan NoSQL



Cách viết CV

HOW TO WRITE A CV



Tại sao lại là NoSQL

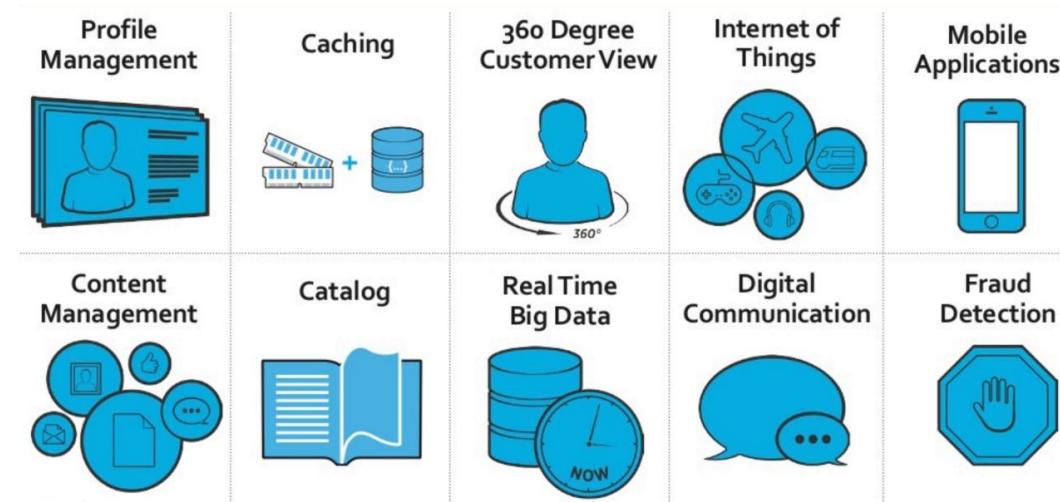
- Các ứng dụng web có nhu cầu khác nhau • Khả năng mở rộng theo chiều ngang - giảm chi phí • Phân bổ theo địa lý • Tính đàn hồi • Không cần lược đồ, lược đồ linh hoạt cho dữ liệu bán cấu trúc • Dễ dàng hơn cho các nhà phát triển • Lưu trữ dữ liệu không đồng nhất • Khả dụng cao/Phục hồi sau thảm họa
- Ứng dụng web không phải lúc nào cũng cần • Giao dịch • Tính nhất quán mạnh mẽ • Truy vấn phức tạp

SQL so với NoSQL

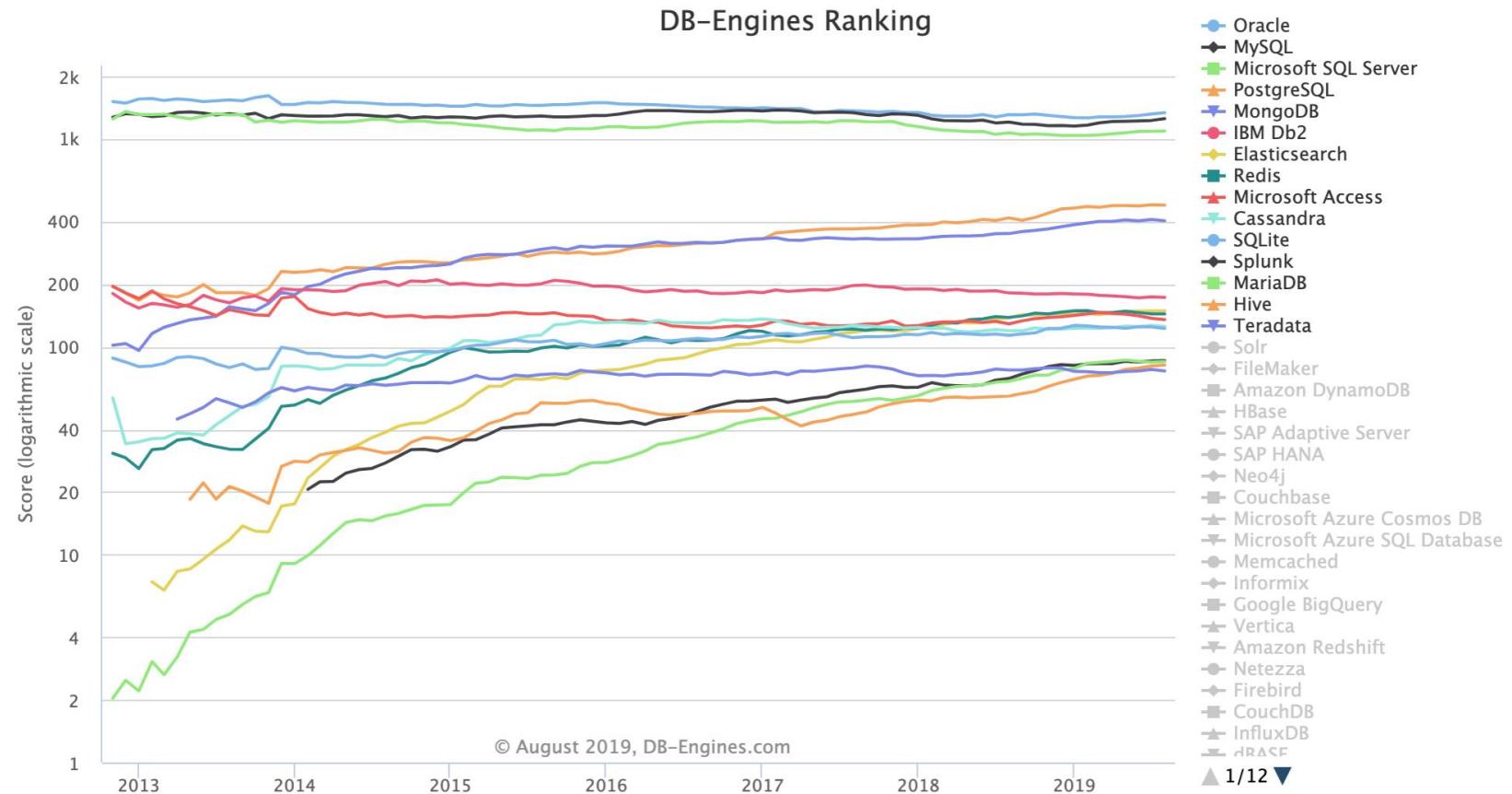
SQL	Không có SQL
Gigabyte sang Terabyte	Petabyte (1kTB) đến Exabyte (1kPB) đến Zetabyte (1kEB)
Tập trung	Phân phối
Có cấu trúc	Bán cấu trúc và Không cấu trúc
Ngôn ngữ truy vấn có cấu trúc	Không có ngôn ngữ truy vấn khai báo
Mô hình dữ liệu ổn định	Sơ đồ ít hơn
Mỗi quan hệ phức tạp	Mỗi quan hệ ít phức tạp hơn
Tính chất ACID	Sự nhất quán cuối cùng
Giao dịch được ưu tiên	Tính khả dụng cao, khả năng mở rộng cao
Tham gia bảng	Cấu trúc nhúng

Các trường hợp sử dụng NoSQL

- Khối lượng dữ liệu lớn ở quy mô lớn (Big volume)
 - Google, Amazon, Yahoo, Facebook - 10-100K máy chủ
- Khối lượng truy vấn cực lớn (Tốc độ lớn)
- Tính khả dụng cao
- Linh hoạt, tiến hóa lược đồ

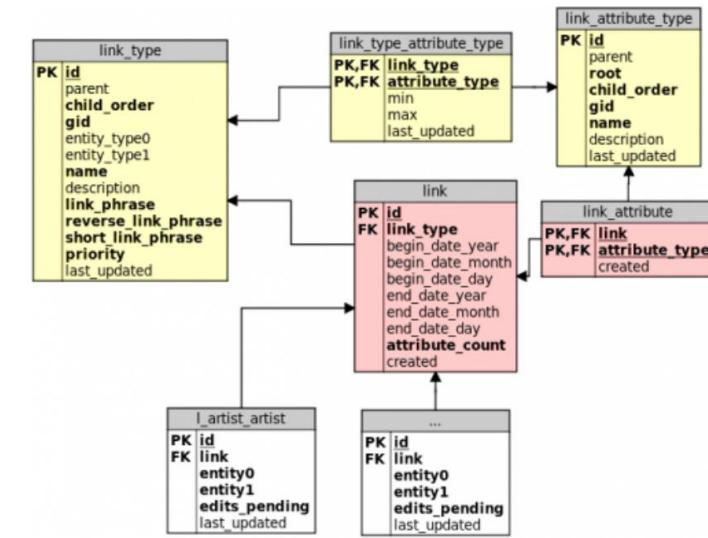


Xếp hạng động cơ DB theo mức độ phổ biến (2019)



Mô hình dữ liệu quan hệ được xem xét lại

- Dữ liệu thường được lưu trữ theo từng hàng cách thức (hàng lưu trữ)
- Ngôn ngữ truy vấn chuẩn hóa (SQL)
- Mô hình dữ liệu được xác định trước khi bạn thêm dữ liệu
- Kết hợp dữ liệu từ nhiều bảng
 - Kết quả là bảng
- Ưu điểm: Giao dịch ACID trưởng thành với các biện pháp kiểm soát bảo mật chi tiết, được sử dụng rộng rãi
- Nhược điểm: Yêu cầu mô hình hóa dữ liệu trước, không có tỷ lệ tốt



Oracle, MySQL, PostgreSQL,
Máy chủ Microsoft SQL, IBM
DB/2

Mô hình dữ liệu khóa/giá trị

- Giao diện khóa/giá trị đơn giản
 - GET, PUT, DELETE
- Giá trị có thể chứa bất kỳ loại dữ liệu nào
 - Siêu nhanh và dễ mở rộng quy mô (không cần nối)
 - Ví dụ
 - Berkley DB, Memcache, DynamoDB, Redis, Riak

key	value
firstName	Bugs
lastName	Bunny
location	Earth

PRODUCT	PRICE
WIDGET	\$118
GIZMO	\$88
TRINKET	\$37
THINGAMAJIG	\$18
DOODAD	\$60
TCHOTCHKE	\$999



PRODUCT	PRICE
TRINKET	\$37
THINGAMAJIG	\$18

PRODUCT	PRICE
GIZMO	\$88
DOODAD	\$60

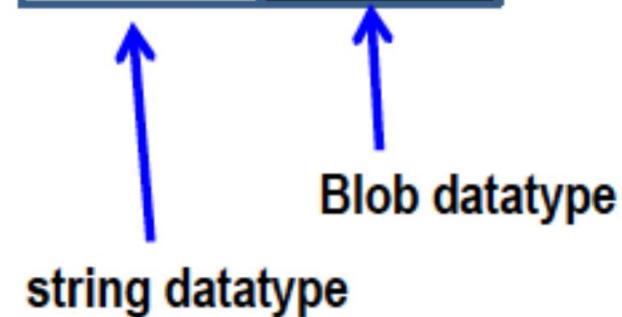
PRODUCT	PRICE
WIDGET	\$118
TCHOTCHKE	\$999

Khóa/giá trị so với bảng

- Một bảng có hai cột và một giao diện đơn giản •
Thêm một
khóa-giá trị • Đôi
với khóa này, hãy cung cấp cho tôi
giá trị • Xóa một khóa

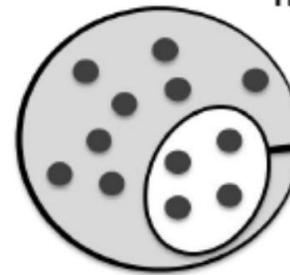


Key	Value



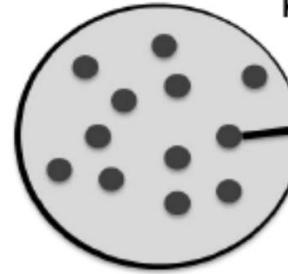
Khóa/giá trị so với mô hình dữ liệu quan hệ

Traditional Relational Model



- Result set based on row values
- Value of rows for large data sets must be indexed
- Values of columns must all have the same data type

Key-Value Store Model



- All queries return a single item
- No indexes on values
- Values may contain any data type

Memcached



- Hệ thống lưu trữ đệm khóa-giá trị trong bộ nhớ nguồn mở • Sử dụng hiệu quả RAM trên nhiều trang web phân tán máy chủ
- Được thiết kế để tăng tốc các ứng dụng web động bằng cách giảm tải cơ sở dữ liệu • Giao diện đơn giản cho bộ nhớ đệm RAM phân tán cao • Thời gian đọc thông thường là 30ms • Được thiết kế để triển khai nhanh chóng, dễ phát triển • API bằng nhiều ngôn ngữ

Đỗ lại

- Kho lưu trữ khóa-giá trị trong bộ nhớ nguồn mở với tùy chọn độ bền
- Tập trung vào việc đọc và ghi tốc độ cao các cấu trúc dữ liệu phổ biến vào RAM
- Cho phép các danh sách, tập hợp và băm đơn giản được lưu trữ trong giá trị và được thao tác
- Nhiều tính năng mà các nhà phát triển thích hết hạn, giao dịch, pub/sub, phân vùng



Amazon DynamoDB

- Kho lưu trữ khóa-giá trị có thể mở rộng
- Sản phẩm tăng trưởng nhanh nhất trong lịch sử của Amazon
- Tập trung vào thông lượng lưu trữ và thời gian đọc và ghi có thể dự đoán được
- Tích hợp mạnh mẽ với S3 và Elastic MapReduce



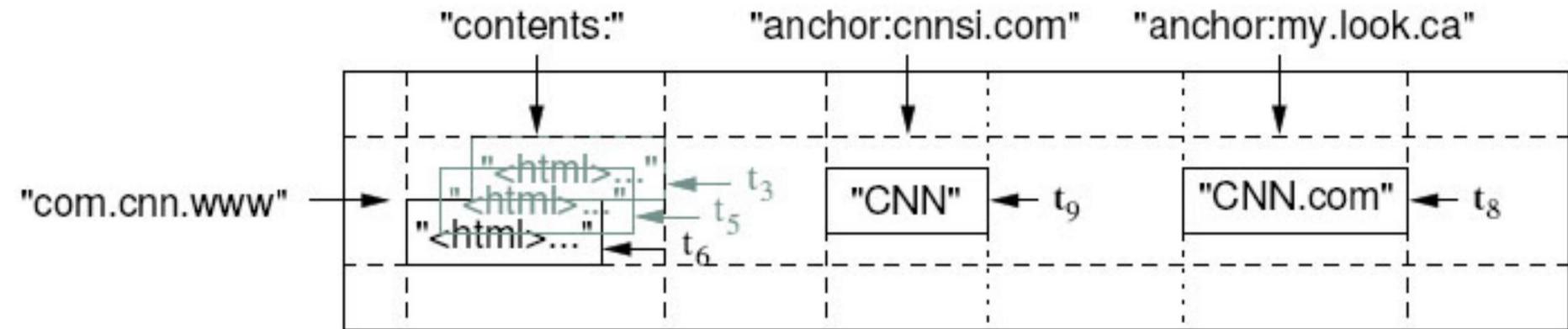
Riak

- Kho lưu trữ khóa-giá trị phân tán nguồn mở với các phiên bản hỗ trợ và thương mại của Basho
- Cơ sở dữ liệu "lấy cảm hứng từ Dynamo"
- Tập trung vào tính khả dụng, khả năng chịu lỗi, tính đơn giản trong vận hành và khả năng mở rộng
- Hỗ trợ sao chép và tự động phân mảnh và cân bằng lại khi xảy ra lỗi
- Hỗ trợ MapReduce, tìm kiếm toàn văn và thứ cấp chỉ số của thẻ giá trị
- Được viết bằng ERLANG



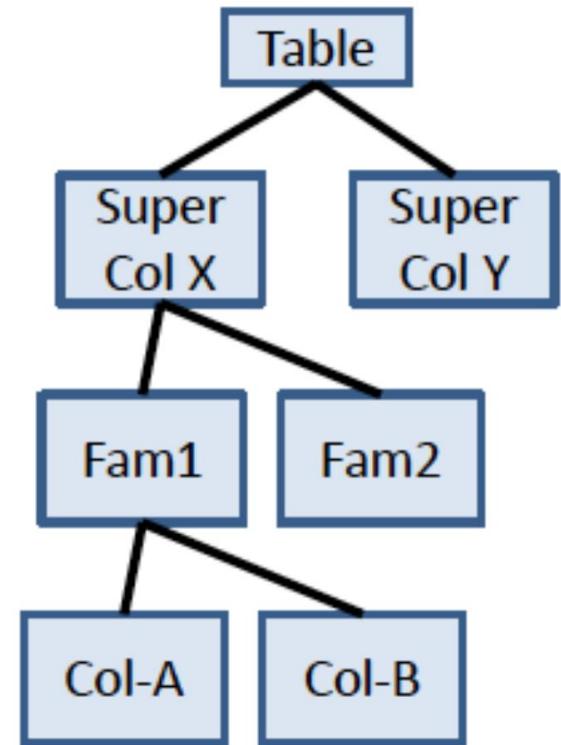
Cửa hàng gia đình cột

- Sơ đồ động, mô hình dữ liệu theo cột • Phân tán, thưa thớt, đa chiều được sắp xếp liên tục bản đồ
- (hàng, cột (họ), dấu thời gian) -> nội dung ô



Họ cột

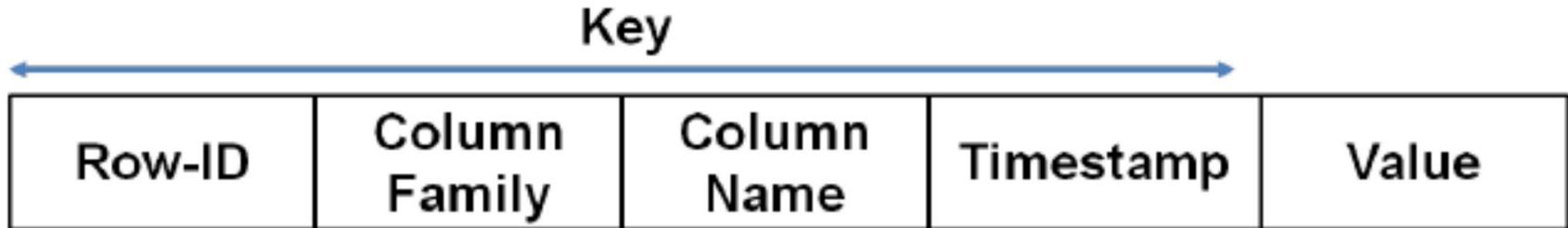
- Nhóm các cột thành "Họ cột"
- Nhóm các họ cột thành "Siêu Cột"
- Có thể truy vấn tất cả các cột có họ hoặc siêu họ
- Dữ liệu tương tự được nhóm lại với nhau để cải thiện tốc độ



Mô hình dữ liệu họ cột so với quan hệ

- Ma trận thưa thớt, bảo toàn cấu trúc bảng
 - Một hàng có thể có hàng triệu cột nhưng có thể rất thưa thớt
- Lưu trữ hàng/cột hỗn hợp
- Số lượng cột có thể mở rộng
 - Các cột mới được chèn vào mà không cần thực hiện "thay đổi bảng"

Key



Row-ID	Column Family	Column Name	Timestamp	Value
--------	---------------	-------------	-----------	-------

Bàn lớn

- ACM TOCS 2008
- Chịu lỗi, bền bỉ
- Có thể mở rộng
 - Hàng ngàn máy chủ
 - Terabyte dữ liệu trong bộ nhớ • Petabyte dữ liệu trên đĩa • Hàng triệu lần đọc/ghi mỗi giây, quét hiệu quả
- Tự quản lý
 - Máy chủ có thể được thêm/xóa một cách linh hoạt
 - Máy chủ điều chỉnh để mätt cân bằng tải

Bigtable: A Distributed Storage System for Structured Data

Full Text:  PDF  [Get this Article](#)

Authors:	Fay Chang	Google, Inc.
	Jeffrey Dean	Google, Inc.
	Sanjay Ghemawat	Google, Inc.
	Wilson C. Hsieh	Google, Inc.
	Deborah A. Wallach	Google, Inc.
	Mike Burrows	Google, Inc.
	Tushar Chandra	Google, Inc.
	Andrew Fikes	Google, Inc.
	Robert E. Gruber	Google, Inc.



Published in:



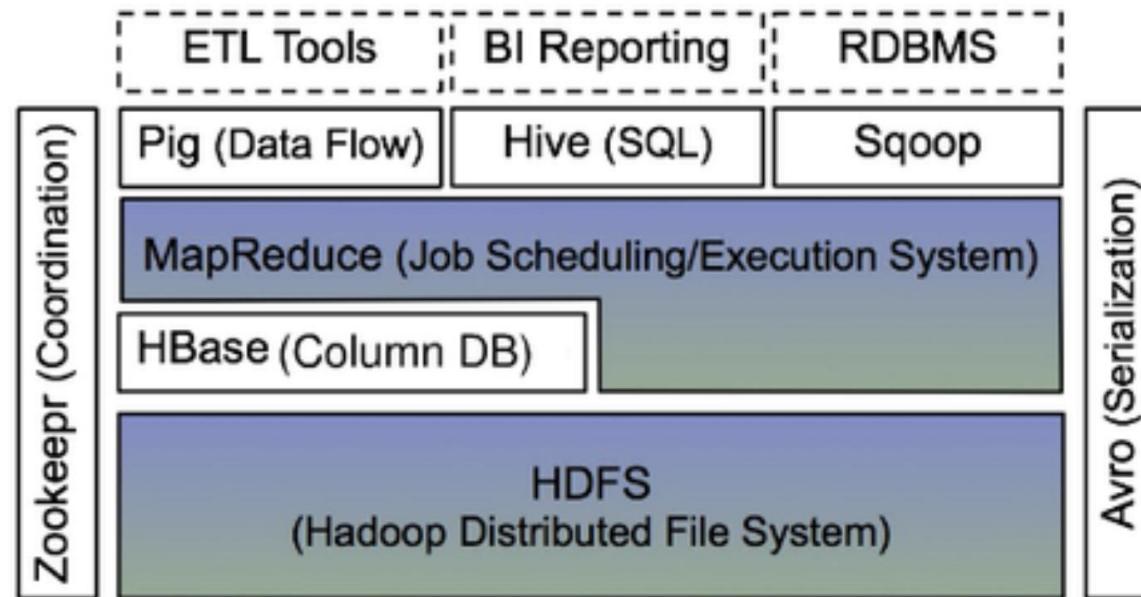
- Journal

ACM Transactions on Computer Systems (TOCS) [TOCS Homepage](#) [archive](#)
 Volume 26 Issue 2, June 2008
 Article No. 4
[ACM New York, NY, USA](#)
[table of contents](#) [doi>10.1145/1365815.1365816](#)

Apache Hbase



- Bigtable mã nguồn mở, được viết bằng JAVA
- Một phần của dự án Apache Hadoop



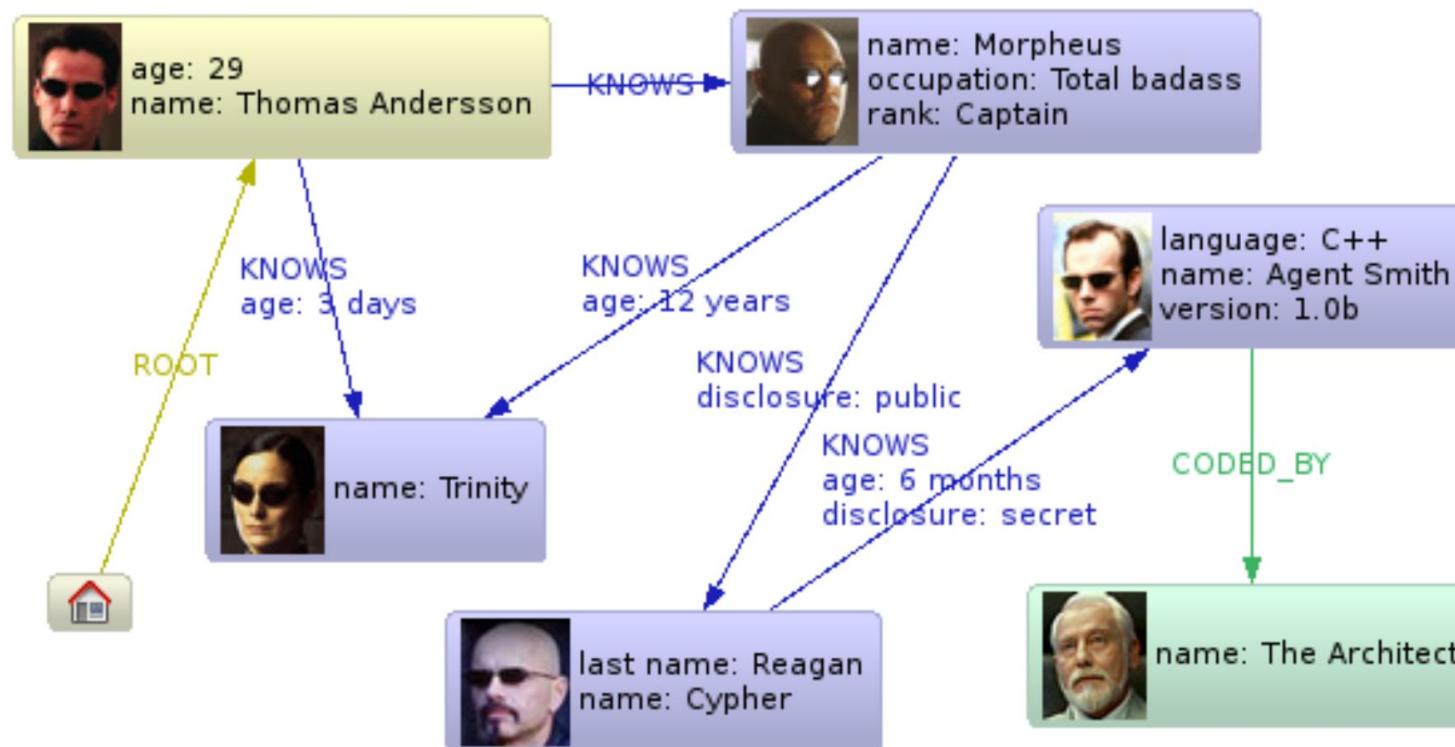
Apache Cassandra

- Cơ sở dữ liệu họ cột mã nguồn mở Apache
- Được hỗ trợ bởi DataStax
- Mô hình phân phối ngang hàng
- Uy tín cao về khả năng mở rộng tuyến tính (hàng triệu lần ghi/giây)
- Được viết bằng Java và hoạt động tốt với HDFS và Bản đồ Giảm



Mô hình dữ liệu đồ thị

- Các trừu tượng cốt lõi: Các nút, Mối quan hệ, Thuộc tính trên cả hai



Lưu trữ cơ sở dữ liệu đồ thị

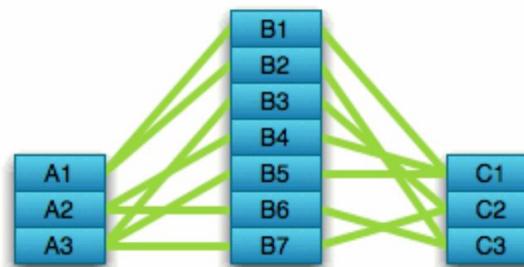
- Một cơ sở dữ liệu lưu trữ dữ liệu trong một cấu trúc đồ thị rõ ràng
- Mỗi nút biết các nút liền kề của nó

Các truy vấn thực sự là các phép duyệt đồ thị

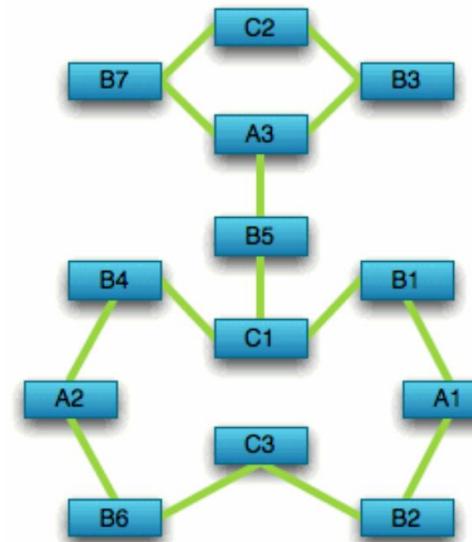


So với Cơ sở dữ liệu quan hệ

Được tối ưu hóa cho tổng hợp



Được tối ưu hóa cho kết nối

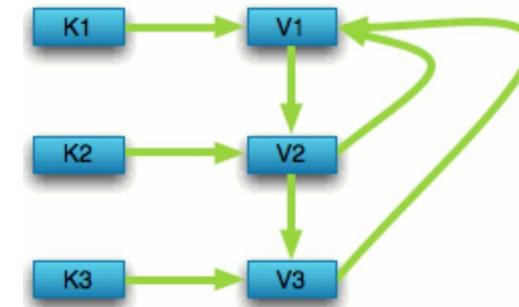


So với các cửa hàng Key Value

Được tối ưu hóa cho việc tra cứu đơn giản



Được tối ưu hóa để truyền dữ liệu được kết nối

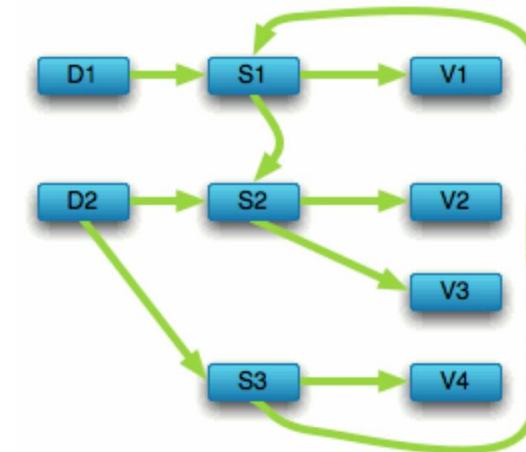


So với kho lưu trữ tài liệu

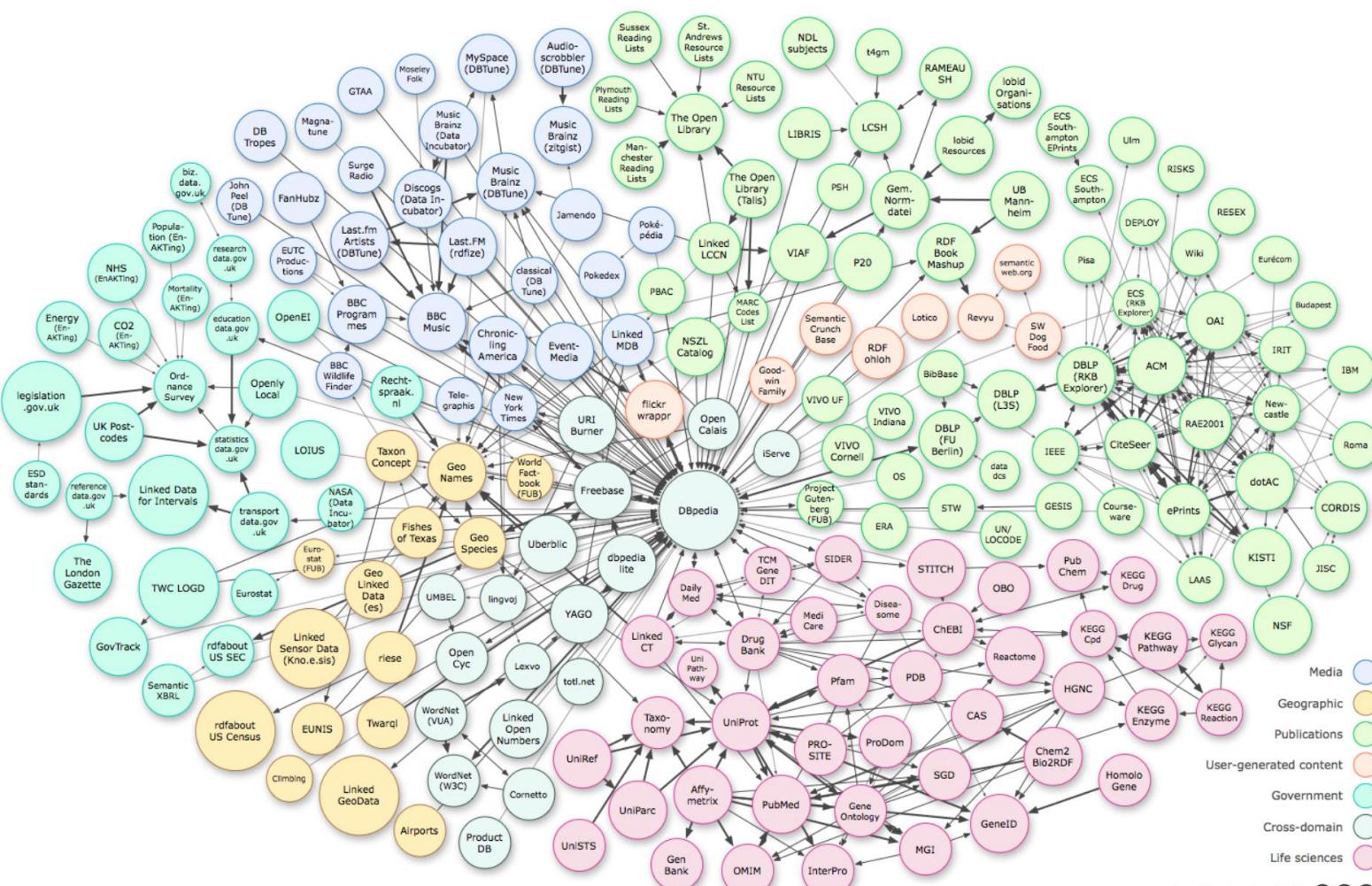
Được tối ưu hóa cho “cây” dữ liệu



Được tối ưu hóa để nhìn thấy khu rừng và cây cối, cành cây và thân cây



Liên kết dữ liệu mở



Neo4j

- Cơ sở dữ liệu đồ thị được thiết kế để các nhà phát triển Java dễ sử dụng
- Dựa trên đĩa (không chỉ RAM)
- AXIT đầy đủ
- Tính khả dụng cao (với Phiên bản doanh nghiệp)
- 32 tỷ nút, 32 tỷ mối quan hệ,
64 Tỷ Bất Động Sản
- Thư viện Java nhúng

- Giao diện lập trình ứng dụng REST



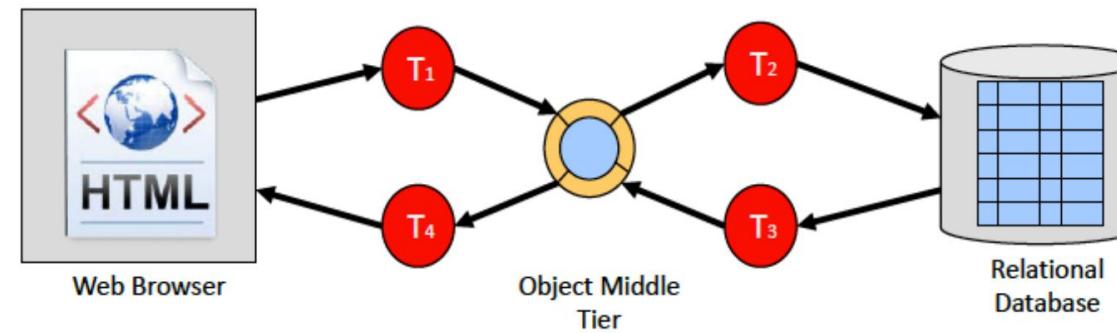
Lưu trữ tài liệu

- Tài liệu, không phải giá trị, không phải bảng
- Định dạng JSON hoặc XML
- Tài liệu được xác định bằng ID • Cho phép lập chỉ mục trên các thuộc tính

```
{  
  person: {  
    first_name: "Peter",  
    last_name: "Peterson",  
    addresses: [  
      {street: "123 Peter St"},  
      {street: "504 Not Peter St"}  
    ],  
  }  
}
```

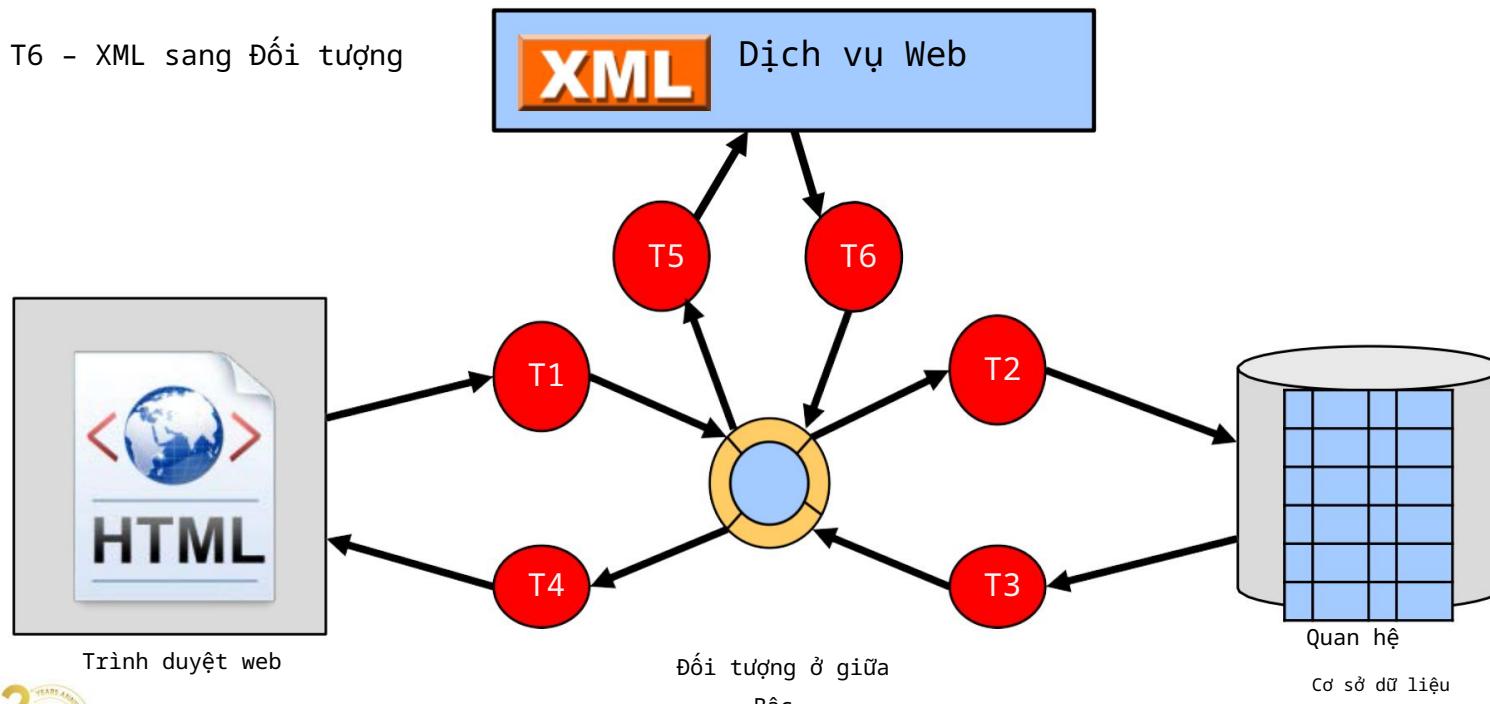
Ánh xạ dữ liệu quan hệ

- T1-HTML thành Đối tượng
- T2-Đối tượng thành Bảng SQL
- T3-Bảng thành Đối tượng
- T4-Đối tượng thành HTML



Dịch vụ Web ở giữa

- T1 - HTML thành Đối tượng Java •
- T2 - Đối tượng Java thành Bảng SQL • T3
- Bảng thành Đối tượng • T4
- Đối tượng thành HTML
- T5 - Đối tượng sang XML
- T6 - XML sang Đối tượng

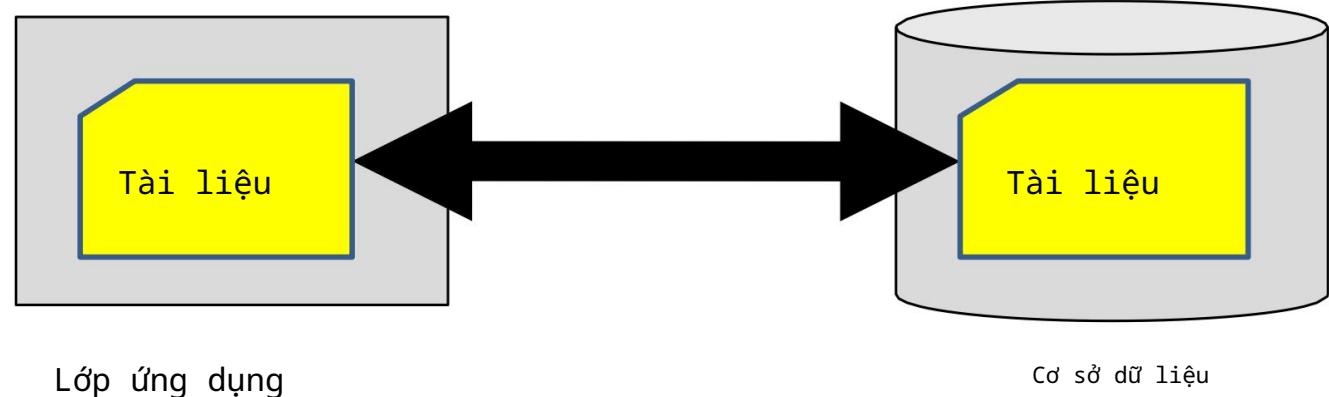


Cuộc thảo luận

- Ánh xạ quan hệ đối tượng đã trở thành một trong những thành phần phức tạp nhất của việc xây dựng các ứng dụng ngày nay
 - Khung Hibernate Java
 - Nhật Bản
- Để tránh sự phức tạp là giữ cho kiến trúc của bạn rất đơn giản

Bản đồ tài liệu

- Tài liệu trong cơ sở dữ liệu
- Tài liệu trong ứng dụng • Không có đối tượng ở tầng trung gian
- Không "băm nhỏ" •
Không cần lắp ráp
lại • Đơn giản!



MongoDB

- Kho dữ liệu JSON nguồn mở được tạo bởi 10gen
- Mô hình mở rộng chủ-tớ
- Cộng đồng nhà phát triển mạnh mẽ
 - Phân mảng tích hợp, tự động
 - Được triển khai bằng C++ với nhiều API (C++, JavaScript, Java, Perl, Python, v.v.)



Kiến trúc MongoDB

- Bộ bản sao •

Các bản sao của dữ liệu
trên mỗi nút

- An toàn dữ liệu
- Tính khả dụng cao •

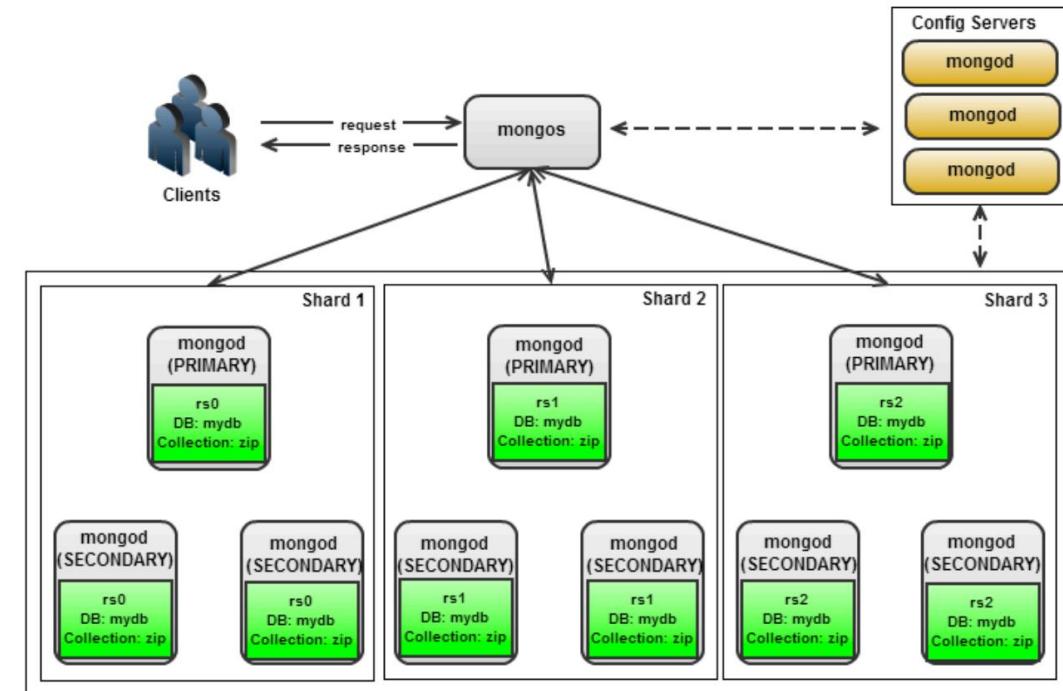
Phục hồi sau thảm họa •

Bảo trì

- Đọc tỷ lệ

- Phân mảnh

- “Phân vùng” của dữ liệu
- Tỷ lệ ngang



Apache CouchDB

- Dự án Apache • Kho dữ liệu JSON nguồn mở • Được viết bằng ERLANG
- API JSON RESTful
- Lập chỉ mục dựa trên B-Tree, phiên bản b-tree theo bóng • Hỗ trợ đầy đủ ACID
- Xem mô hình
- Nén dữ liệu •
Bảo mật



Apache CouchDB™ is a database that uses **JSON** for documents, **JavaScript** for **MapReduce** indexes, and regular **HTTP** for its **API**

Tài liệu tham khảo

- Han, Jing, et al. "Khảo sát về cơ sở dữ liệu NoSQL." 2011 6 Hội nghị quốc tế về điện toán phổ biến và ứng dụng. IEEE, 2011.
- Sivasubramanian, Swaminathan. "Amazon dynamoDB: dịch vụ cơ sở dữ liệu phi quan hệ có khả năng mở rộng liền mạch." Biên bản Hội nghị quốc tế ACM SIGMOD năm 2012 về Quản lý dữ liệu. 2012.
- Chang, Fay, et al. "Bigtable: Hệ thống lưu trữ phân tán cho dữ liệu có cấu trúc." ACM Transactions on Computer Systems (TOCS) 26.2 (2008): 1-26.
- Iordanov, Borislav. "HyperGraphDB: một đồ thị tổng quát cơ sở dữ liệu." Hội nghị quốc tế về quản lý thông tin thời đại web. Springer, Berlin, Heidelberg, 2010.



25
YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Cảm ơn sự
chú ý
của bạn!!!

