

25 YEARS ANNIVERSARY
SICT

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

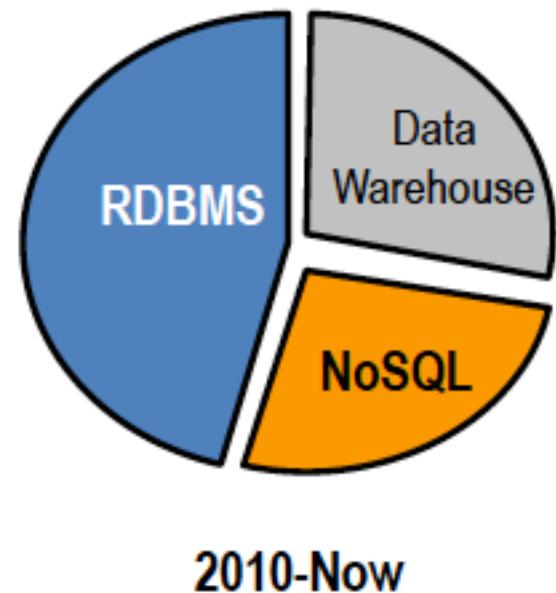
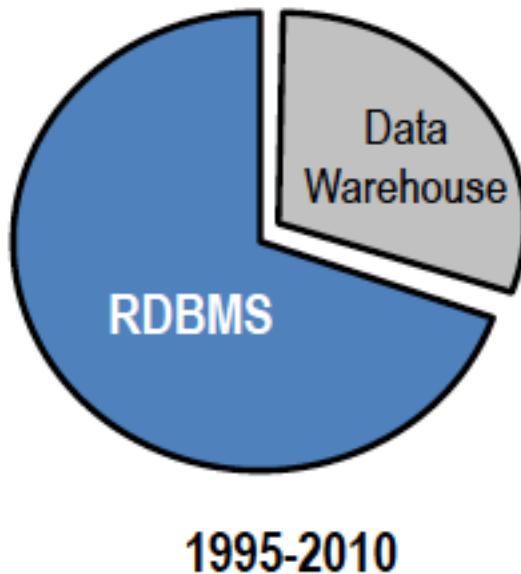
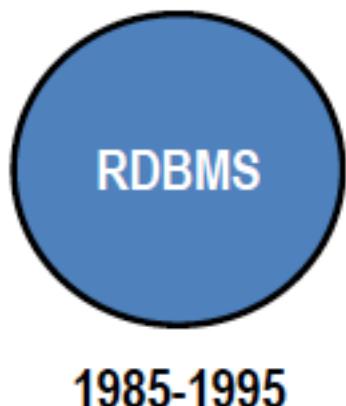


HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

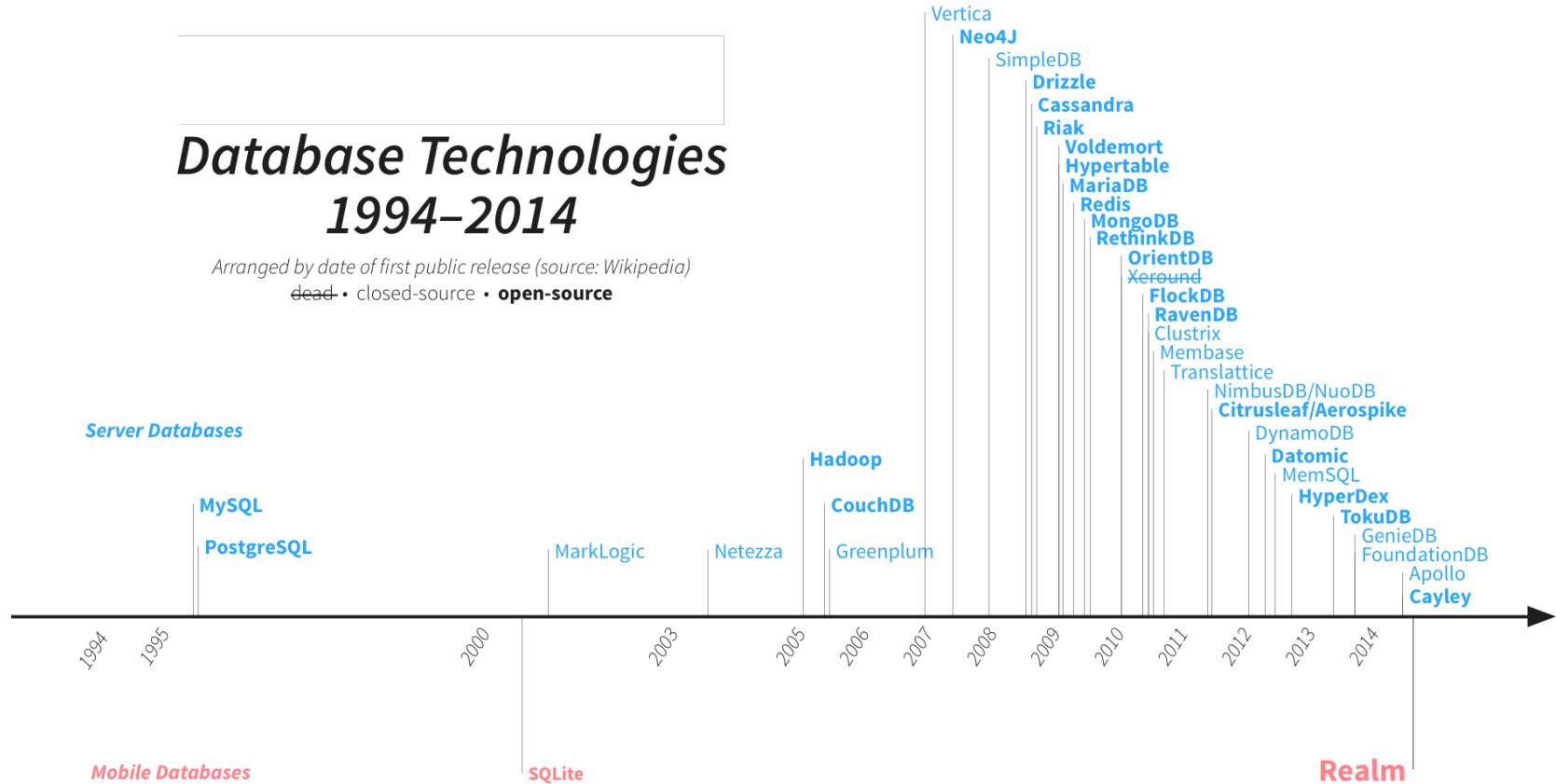
Chương 4

NoSQL - phần 1

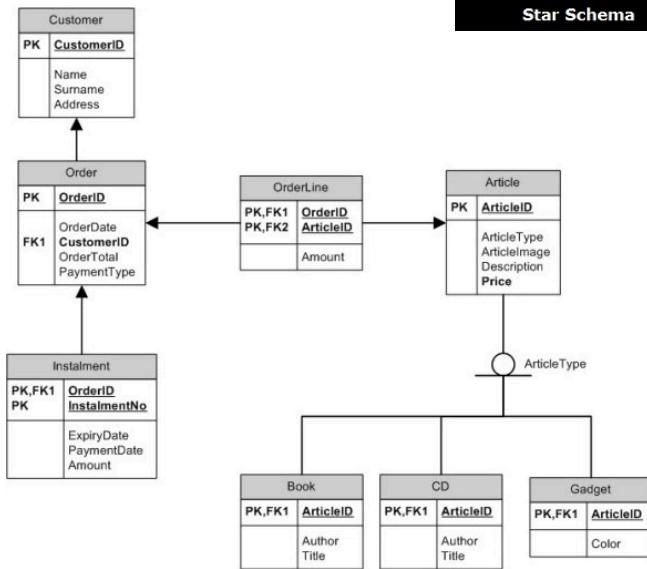
Kỷ nguyên của cơ sở dữ liệu



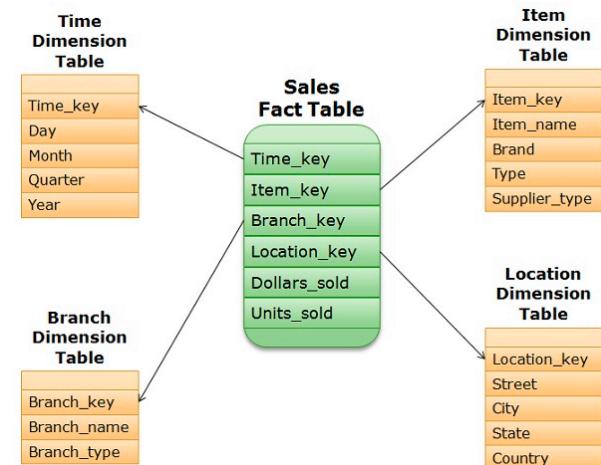
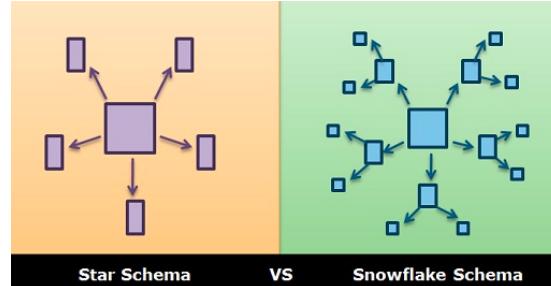
Kỷ nguyên của cơ sở dữ liệu



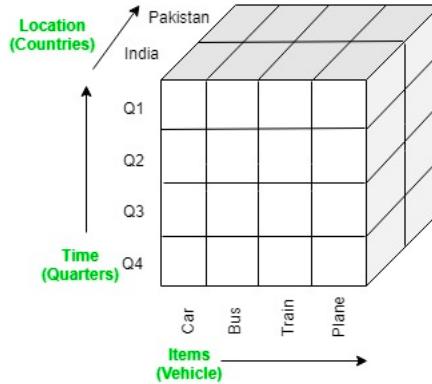
Trước NoSQL



OLTP

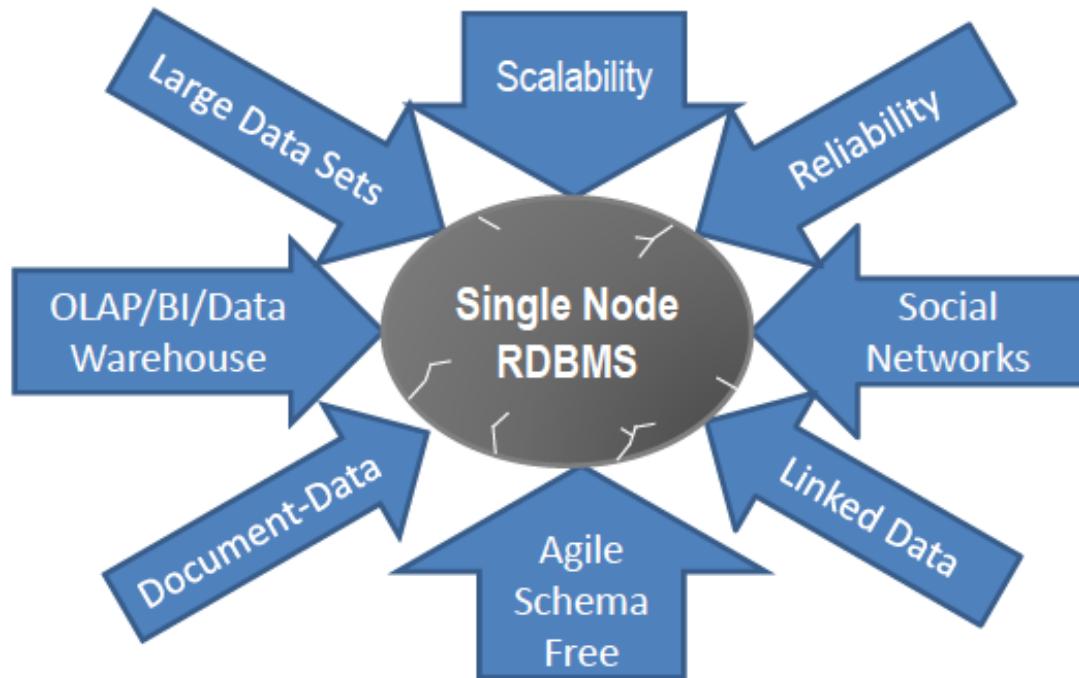


Lược đồ sao



khối OLAP

RDBMS: một kích thước phù hợp với mọi 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)



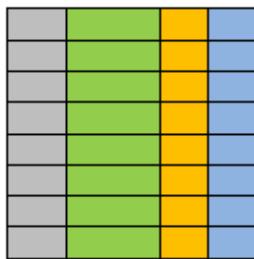
Bibliometrics

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

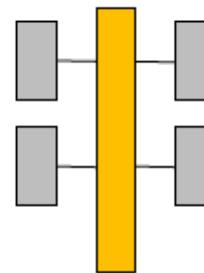
25 năm phát triển DBMS thương mại vừa qua có thể được tóm tắt bằng một cụm từ duy nhất: "một kích thước phù hợp với tất cả". Cụm từ này đề cập đến thực tế là **Kiến trúc DBMS truyền thống (được thiết kế ban đầu và tối ưu hóa để xử lý dữ liệu kinh doanh) đã được sử dụng để hỗ trợ nhiều ứng dụng lấy dữ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

Sau NoSQL

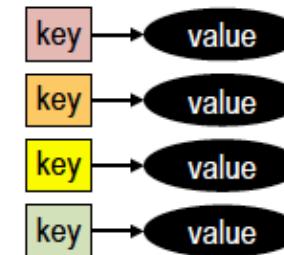
Relational



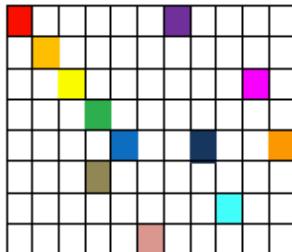
Analytical (OLAP)



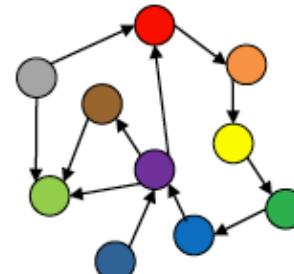
Key-Value



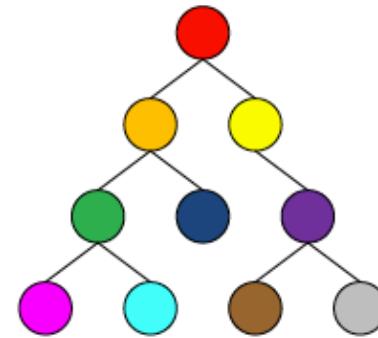
Column-Family



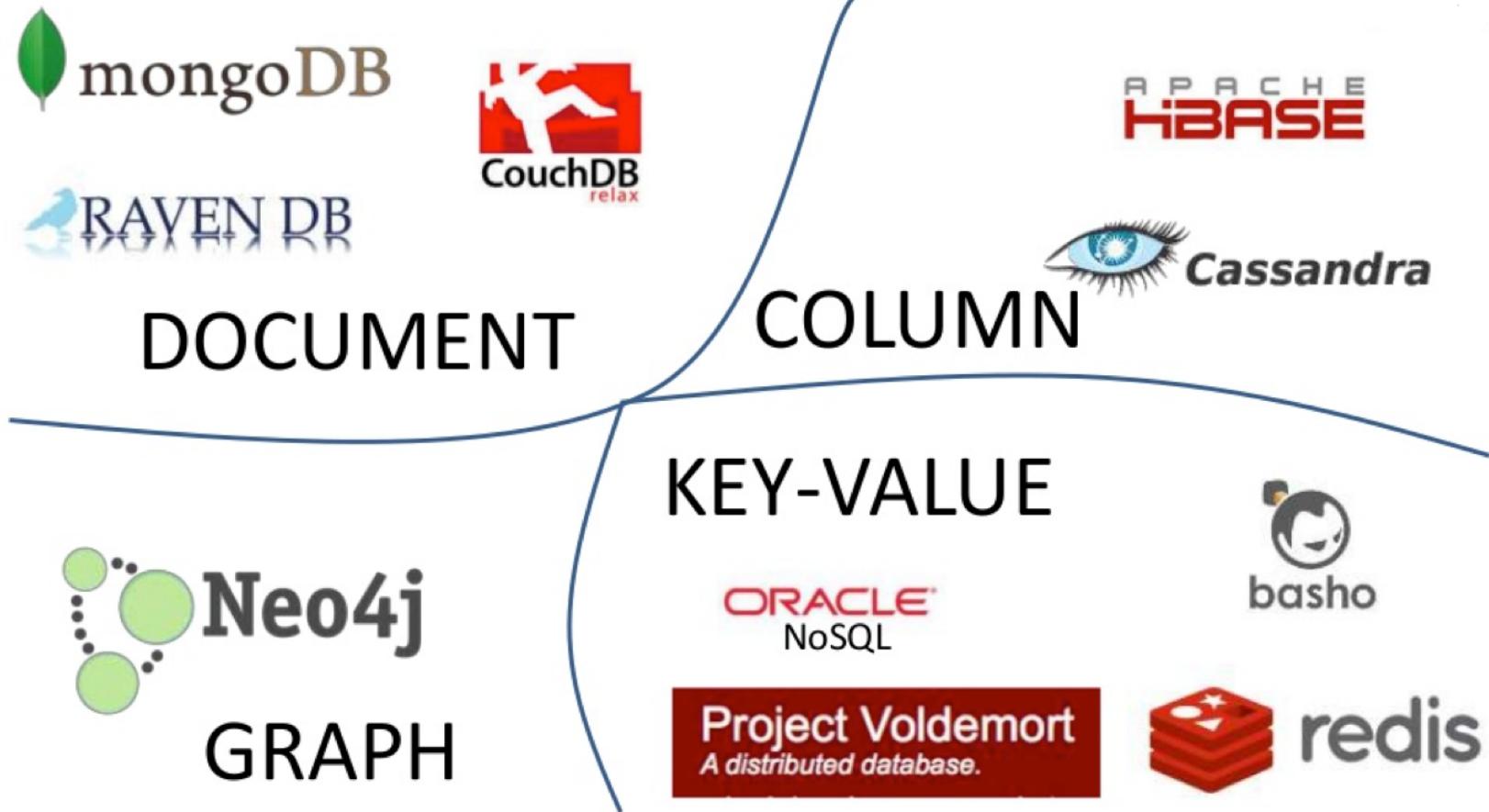
Graph



Document

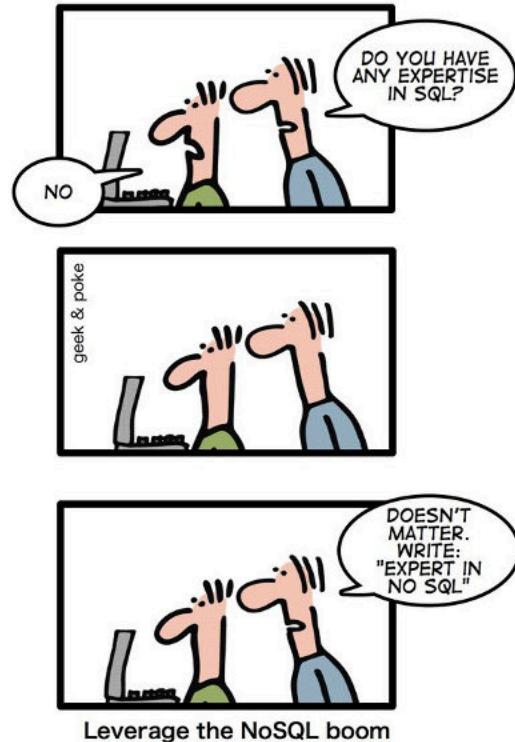


Cánh quan NoSQL



Cách viết CV

HOW TO WRITE A CV



Tại sao NoSQL

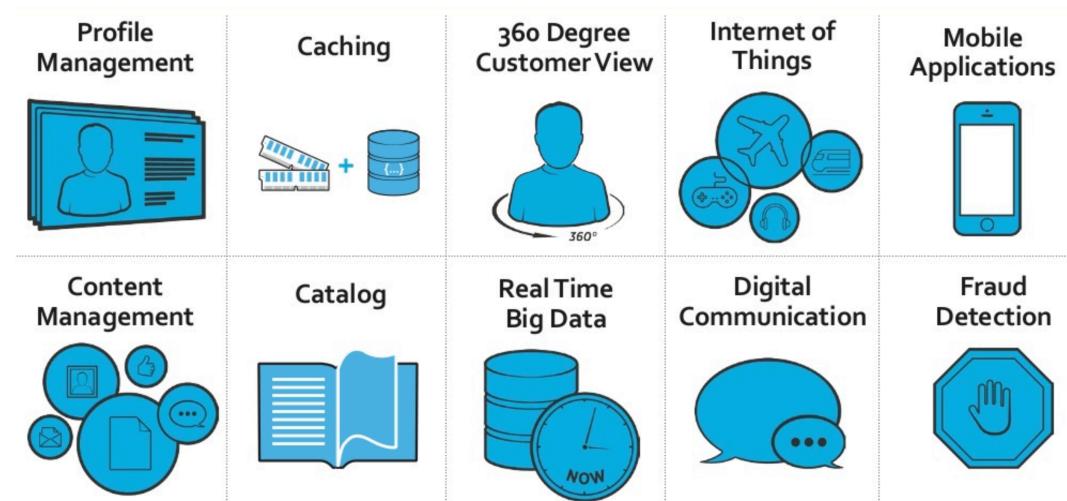
- Ứng dụng web có những nhu cầu khác nhau
 - Khả năng mở rộng theo chiều ngang – giảm chi phí
 - Phân bố theo địa lý
 - Độ đàn hồi
 - Lược đồ ít linh hoạt hơn cho dữ liệu bán cấu trúc
 - Dễ dàng hơn cho nhà phát triển
 - Lưu trữ dữ liệu không đồng nhất
 - Tính sẵn sàng cao/khắc phục thảm họa
- Các ứng dụng web không phải lúc nào cũng cần
 - Giao dịch
 - Tính nhất quán mạnh mẽ
 - Truy vấn phức tạp

SQL so với NoSQL

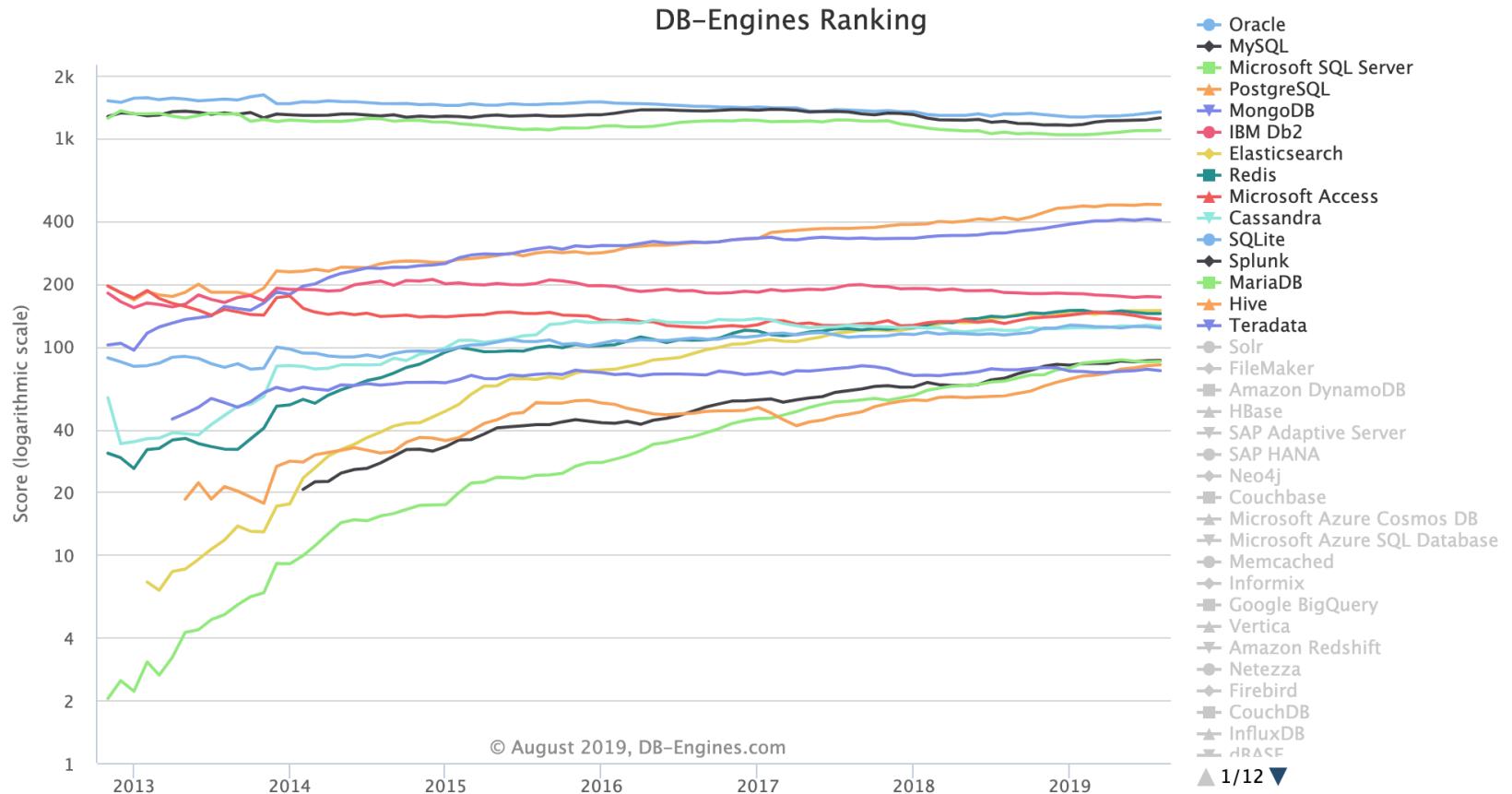
SQL	NoSQL
Gigabyte sang Terabyte	Petabyte(1kTB) tới Exabyte(1kPB) tới Zetabyte(1kEB)
Tập trung	phân phối
Có cấu trúc	Bán cấu trúc và không cấu trúc
Structured Query Language	Không có ngôn ngữ truy vấn khai báo
Mô hình dữ liệu ổn định	Lược đồ ít hơn
Mỗi quan hệ phức tạp	Mỗi quan hệ ít phức tạp hơn
Thuộc tính axit	Tính nhất quán cuối cùng
Ưu tiên giao dịch	Tính sẵn sàng cao, khả năng mở rộng cao
Tham gia bảng	Cấu trúc nhúng

Các trường hợp sử dụng NoSQL

- Khối lượng dữ liệu khổng lồ trên quy mô lớn (Big Volume)
 - Google, Amazon, Yahoo, Facebook – 10-100K máy chủ
- Khối lượng truy vấn cực lớn (Tốc độ lớn)
- Tính sẵn sàng cao
- Phát triển lược đồ linh hoạt

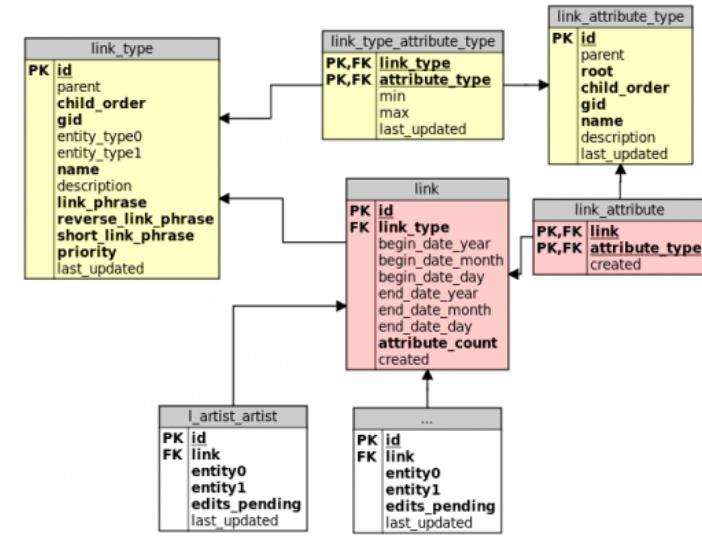


Xếp hạng công cụ DB theo mức độ phổ biến của chúng (2019)



Xem lại mô hình dữ liệu quan hệ

- Dữ liệu thường được lưu trữ theo từng hàng (row store)
- Ngôn ngữ truy vấn được chuẩn hóa (SQL)
- Mô hình dữ liệu được xác định **trước** bạn thêm dữ liệu
- Tham gia hợp nhất dữ liệu từ nhiều bảng
 - Kết quả là bảng
- **Ưu điểm:** Giao dịch ACID hoàn thiện với các biện pháp kiểm soát bảo mật tinh tế, được sử dụng rộng rãi
- **Nhược điểm:** Yêu cầu lập mô hình dữ liệu trước, không có quy mô tốt



Oracle, MySQL, PostgreSQL,
Microsoft SQL Server, IBM DB/
2

Mô hình dữ liệu khóa/giá trị

- Giao diện khóa/giá trị đơn giản
 - NHẬN, ĐẶT, XÓA
- Giá trị có thể chứa bất kỳ loại dữ liệu nào
- Mở rộng quy mô siêu nhanh và dễ dàng (không cần tham gia)
- 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 giao diện đơn giản
 - Thêm khóa-giá trị
 - Đổi với khóa này, hãy cho tôi giá trị
 - Xóa một phím



Key	Value

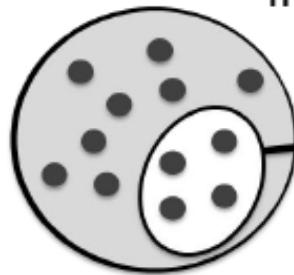


string datatype

Blob datatype

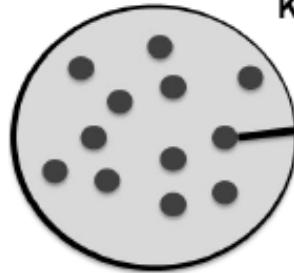
Khóa/giá trị so với mô hình dữ liệu quan hệ

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

Làm lại

- Kho lưu trữ khóa-giá trị trong bộ nhớ nguồn mở với độ bền tùy chọn
- Tập trung vào việc đọc và ghi tốc độ cao các cấu trúc dữ liệu phổ biến vào RAM
- Cho phép lưu trữ các danh sách, bộ và hàm băm đơn giản trong giá trị và được thao tác
- Nhiều tính năng mà nhà phát triển như 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 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



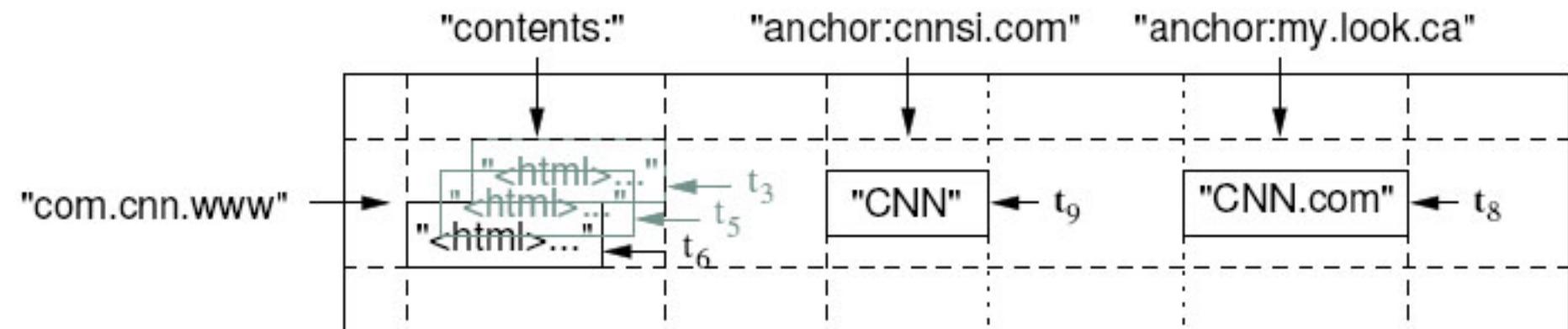
Riak

- Kho lưu trữ khóa-giá trị được phân phối nguồn mở với các phiên bản thương mại và hỗ trợ của Basho
- Cơ sở dữ liệu "lấy cảm hứng từ Dynamo"
- Tập trung vào tính sẵn sàng, khả năng chịu lỗi, tính đơn giản trong vận hành và khả năng mở rộng
- Hỗ trợ sao chép, tự động phân chia và cân bằng lại khi có lỗi
- Hỗ trợ MapReduce, tìm kiếm toàn văn và chỉ mục phụ của thẻ giá trị
- Viết bằng ERLANG



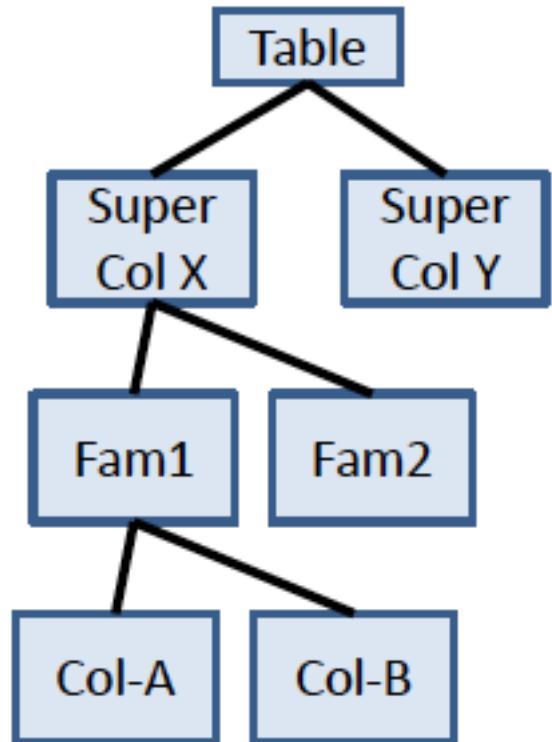
Cửa hàng gia đình cột

- Lược đồ động, mô hình dữ liệu hướng cột
- Bản đồ sắp xếp đa chiều thưa thớt, phân tán liên tục
- (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 họ siêu
- 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, giữ nguyên cấu trúc bảng
 - Một hàng có thể có hàng triệu cột nhưng có thể rất thưa thớt
- Lưu trữ hàng/cột kết hợp
- Số lượng cột có thể mở rộng
 - Các cột mới được chèn mà không cần thực hiện "thay đổi bảng"

Key

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

Cái bàn lớn

- ACM TOCS 2008
- Chịu lỗi, bền bỉ
- Có thể mở rộng
 - Hàng ngàn máy chủ
 - Hàng 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 mất 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:

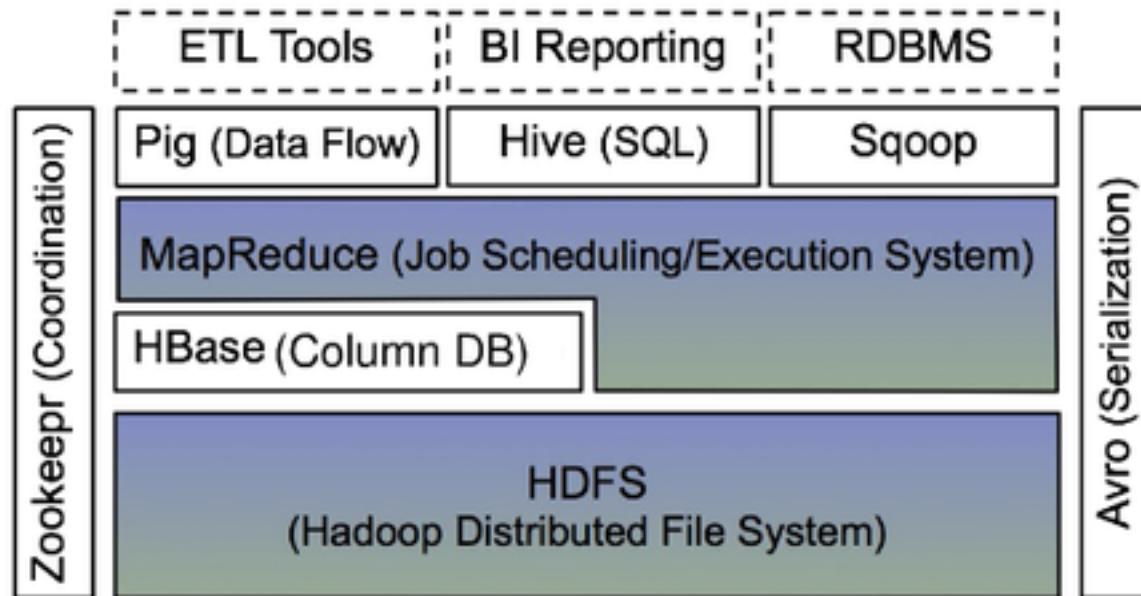


- Journal
ACM Transactions on Computer Systems (TOCS) [TOCS Homepage](#) [archive](#)
Volume 26 Issue 2, June 2008
Article No. 4
[ACM New York, NY, USA](#)
[table of contents](#) doi>[10.1145/1365815.1365816](https://doi.org/10.1145/1365815.1365816)

Cơ sở Apache

APACHE
HBASE

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



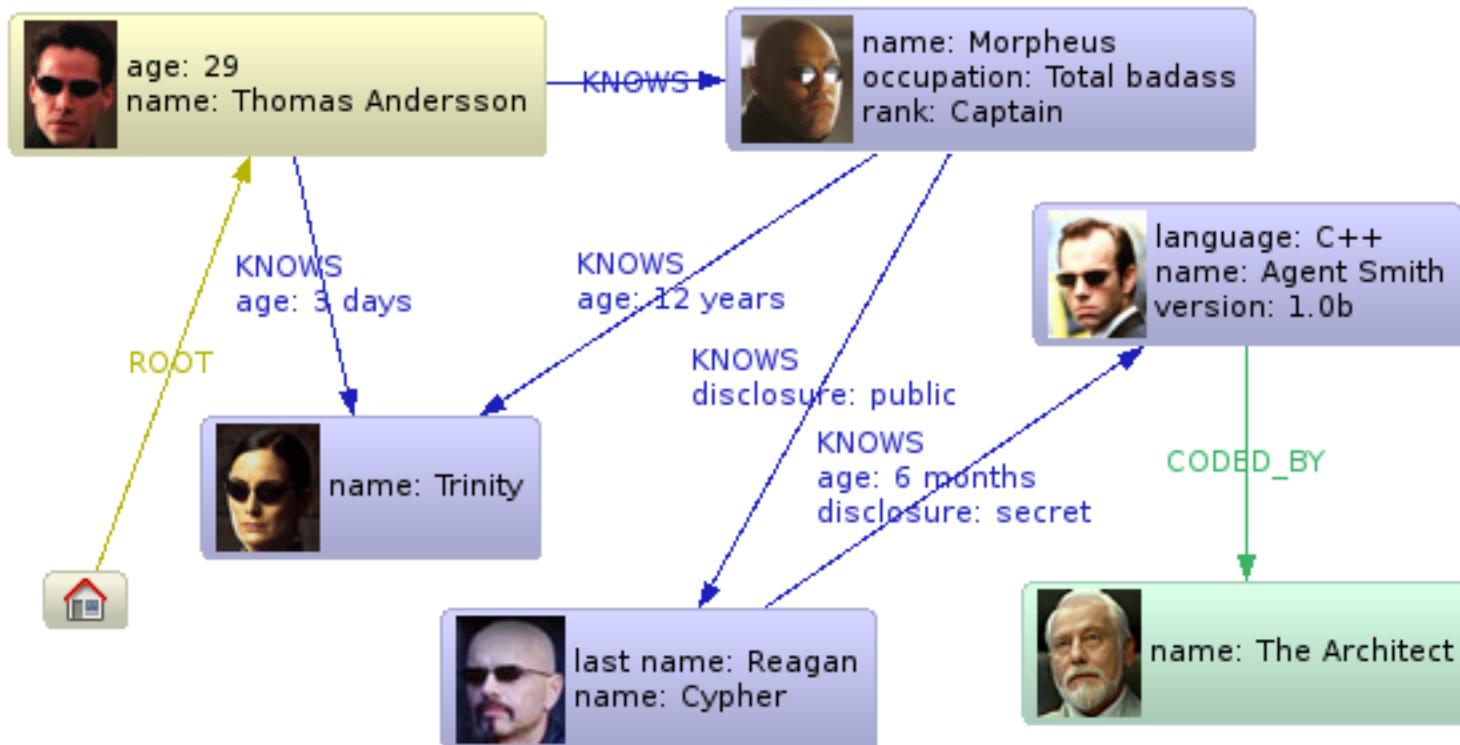
Apache Cassandra

- Cơ sở dữ liệu họ cột mã nguồn mở Apache
- Được hỗ trợ bởi DataStax
- Mô hình phân phối ngang hàng
- Danh tiếng mạnh mẽ về quy mô tuyến tính (hàng triệu lần ghi/giây)
- Được viết bằng Java và hoạt động tốt với HDFS và MapReduce



Mô hình dữ liệu đồ thị

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



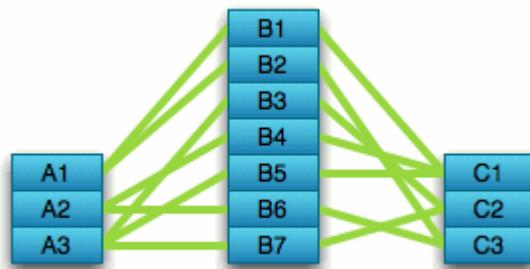
Kho cơ sở dữ liệu đồ thị

- Cơ sở dữ liệu lưu trữ dữ liệu theo cấu trúc đồ thị rõ ràng
- Mỗi nút biết các nút lân cận của nó
- Truy vấn thực sự là việc duyệt đồ thị

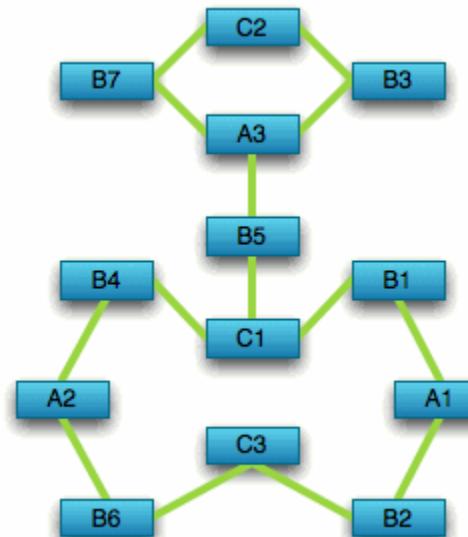


So với cơ sở dữ liệu quan hệ

Tối ưu hóa để tổng hợp



Tối ưu hóa cho kết nối

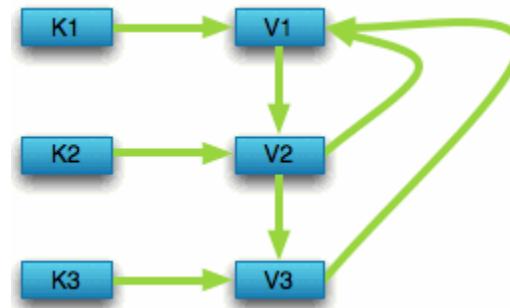


So với các cửa hàng giá trị chính

Tối ưu hóa cho việc tra cứu đơn giản



Tối ưu hóa để duyệt dữ liệu được kết nối

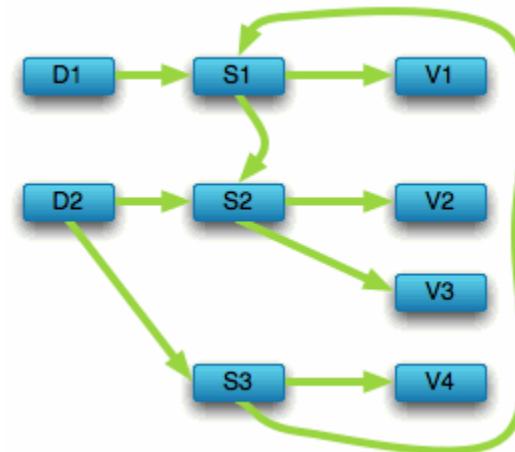


So với các cửa hàng tài liệu

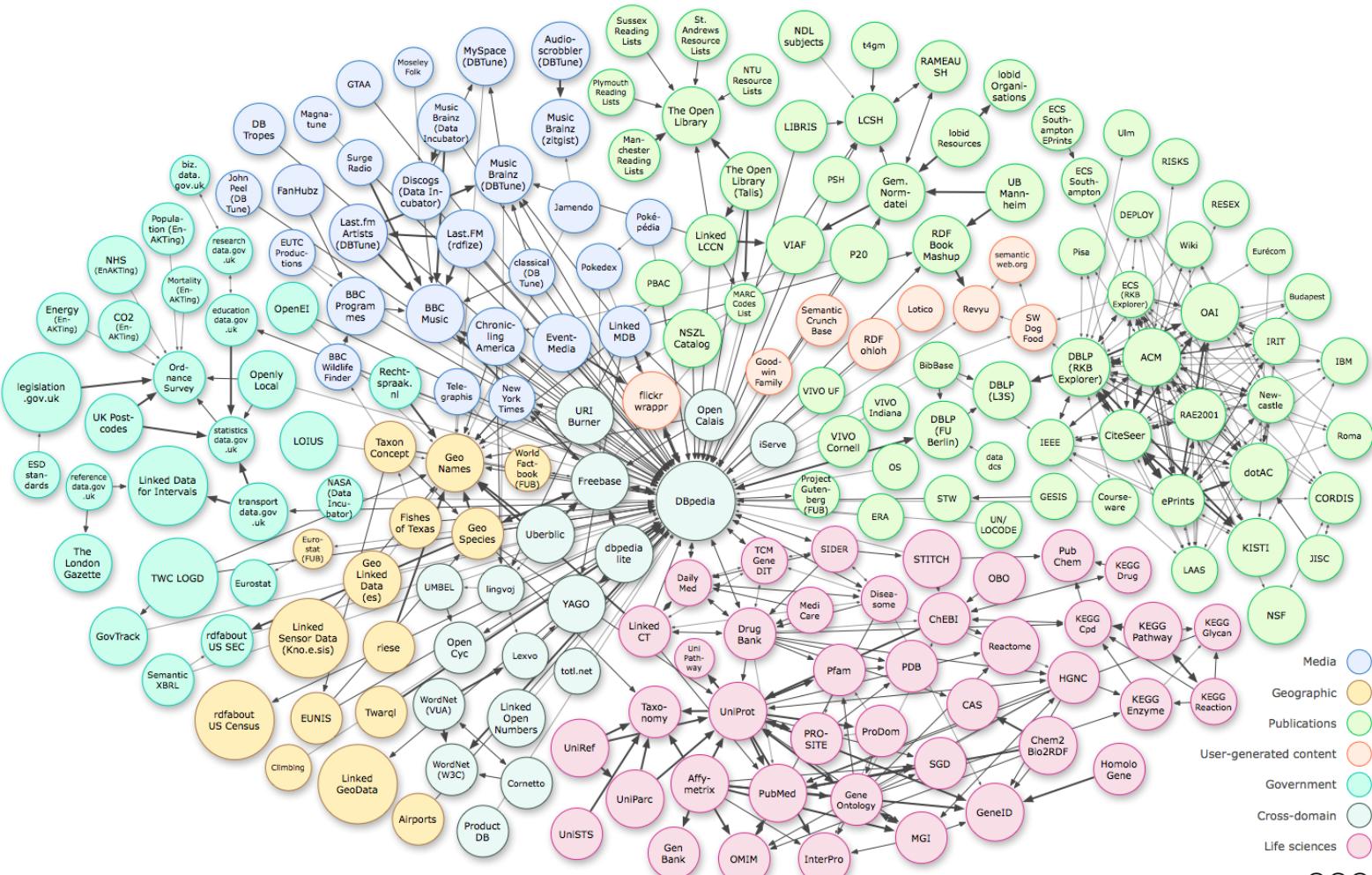
Tối ưu hóa cho “cây” dữ liệu



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



Liên kết dữ liệu mở



As of September 2010



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 sẵn sàng cao (với Phiên bản doanh nghiệp)
- 32 tỷ nút, 32 tỷ mối quan hệ, 64 tỷ thuộc tính
- Thư viện java nhúng
- API REST



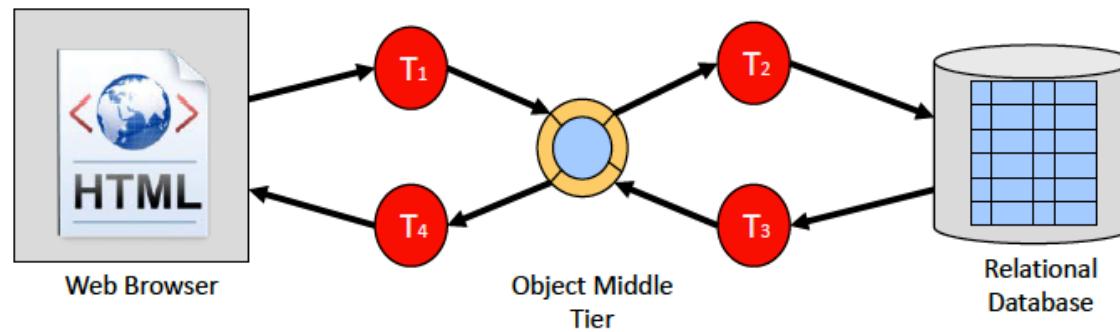
Kho tài liệu

- 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 thuộc tính

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

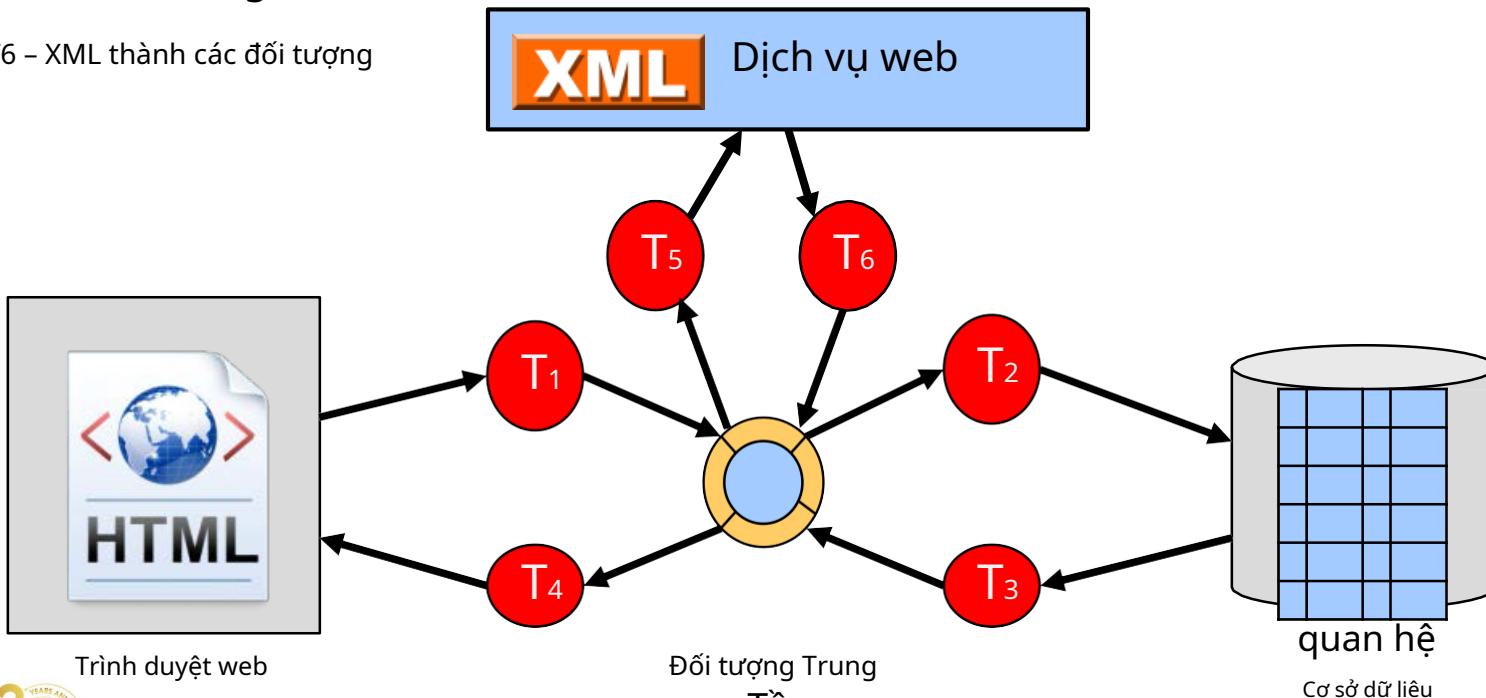
Ánh xạ dữ liệu quan hệ

- T1-HTML thành các đối tượng
- T2-Đối tượng vào bảng SQL
- T3-Bảng thành đối tượng
- T4-Đối tượng thành HTML



Dịch vụ web ở giữa

- T1 – HTML thành các đối tượng Java
- T2 – Chuyển các đối tượng Java vào bảng SQL
- T3 – Biến bảng thành đối tượng
- T4 – Chuyển đổi tượng thành HTML
- T5 – Đổi tượng XML
- T6 – XML thành các đối tượng

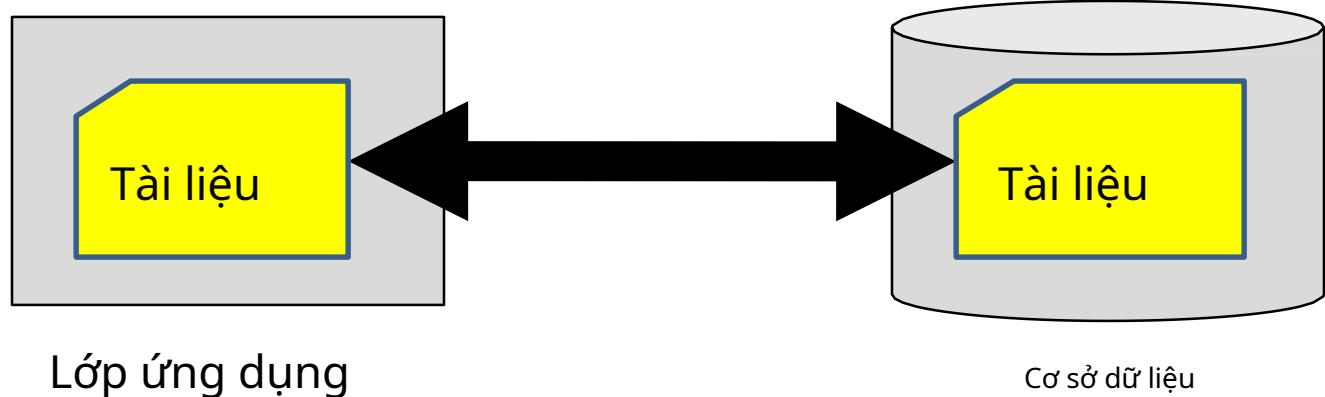


Cuộc thảo luận

- Ánh xạ quan hệ-đối tượng đã trở thành một trong những thành phần phức tạp nhất của ứng dụng xây dựng hiện nay
 - Khung ngẫu đồng Java
 - JPA
- Để tránh sự phức tạp là giữ cho kiến trúc của bạn thật đơn giản

Ánh xạ tài liệu

- Tài liệu trong cơ sở dữ liệu
- Tài liệu trong đơn
- Không có đối tượng ở tầng giữa
- Không "cắt nhỏ"
- Không cần lắp ráp lại
- Đơn giản!



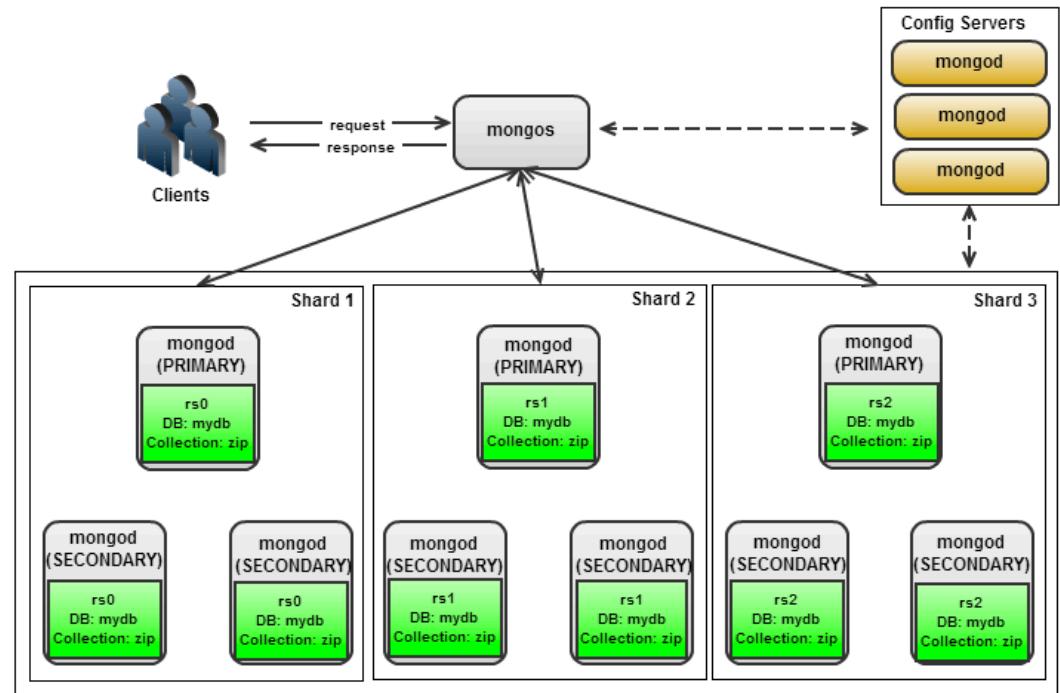
MongoDB

- Kho lưu trữ dữ liệu JSON mã nguồn mở được tạo bởi 10gen
- Mô hình mở rộng quy mô chủ-nô
- Cộng đồng nhà phát triển mạnh mẽ
- Sharding 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
 - Bản sao dữ liệu trên mỗi nút
 - An toàn dữ liệu
 - Tính sẵn sàng cao
 - Khắc phục thảm họa
 - BẢO TRÌ
 - Đọc tỷ lệ
- Phân mảnh
 - “Phân vùng” dữ liệu
 - Tỉ lệ ngang



Apache CouchDB

- Dự án Apache
- Kho dữ liệu JSON nguồn mở
- Viết bằng ERLANG
- API JSON đầy đủ
- Lập chỉ mục dựa trên B-Tree, theo dõi phiên bản b-tree
- ACID được hỗ trợ đầy đủ
- Xem mô hình
- Nén dữ liệu
- Bảo vệ



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

Người giới thiệu

- Han, Jing và cộng sự. "Khảo sát trên cơ sở dữ liệu NoSQL." *Hội nghị quốc tế lần thứ 6 về tính toán và ứng dụng phổ biến*. IEEE, 2011.
- Sivasubramanian, Swaminathan. "Amazon dynamoDB: cơ sở dữ liệu phi quan hệ có khả năng mở rộng liền mạch dịch vụ." *Kỷ yếu Hội nghị Quốc tế ACM SIGMOD 2012 về Quản lý Dữ liệu*. 2012.
- Chang, Fay và cộng sự. "Bigtable: Một hệ thống lưu trữ phân tán cho dữ liệu có cấu trúc." *Giao dịch ACM trên hệ thống máy tính (TOCS)*26.2 (2008): 1-26.
- Iordanov, Borislav. "HyperGraphDB: cơ sở dữ liệu đồ thị tổng quát." *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
cho bạn
chú ý!!!

