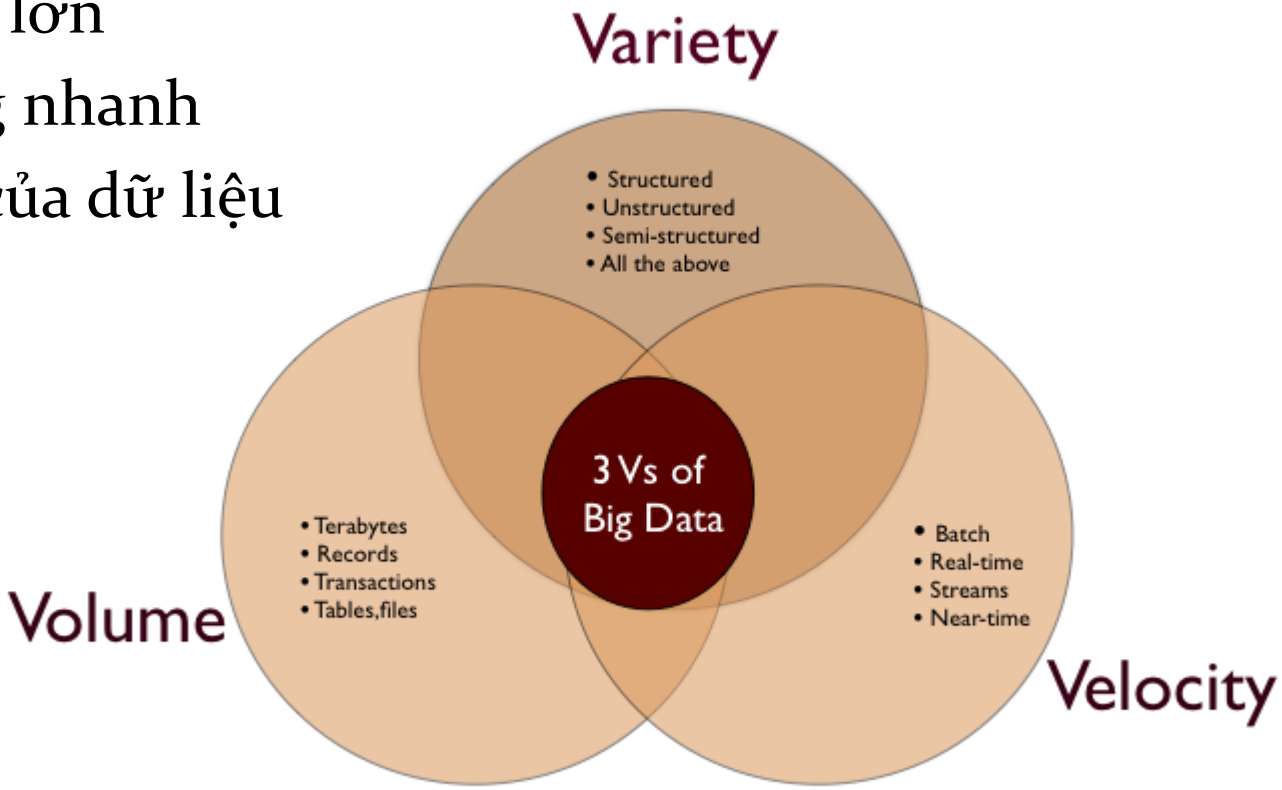
Chương 2. Tổng quan về Cơ sở dữ liệu NoSQL

Nội dung

- Giới thiệu
- NoSQL là gì?
- Đặc trưng của CSDL NoSQL
- Một số khái niệm
- Các loại CSDL NoSQL
- Ưu và nhược điểm của CSDL NoSQL

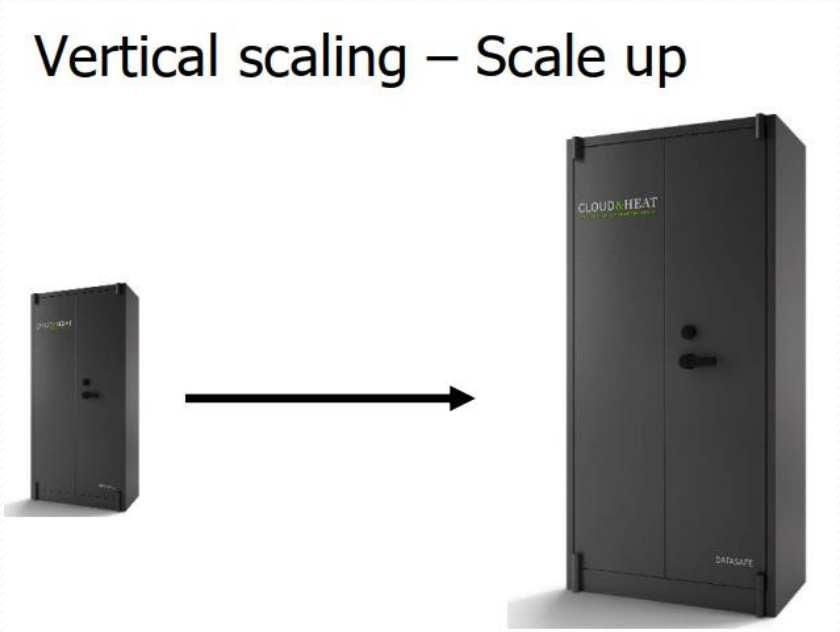
Giới thiệu

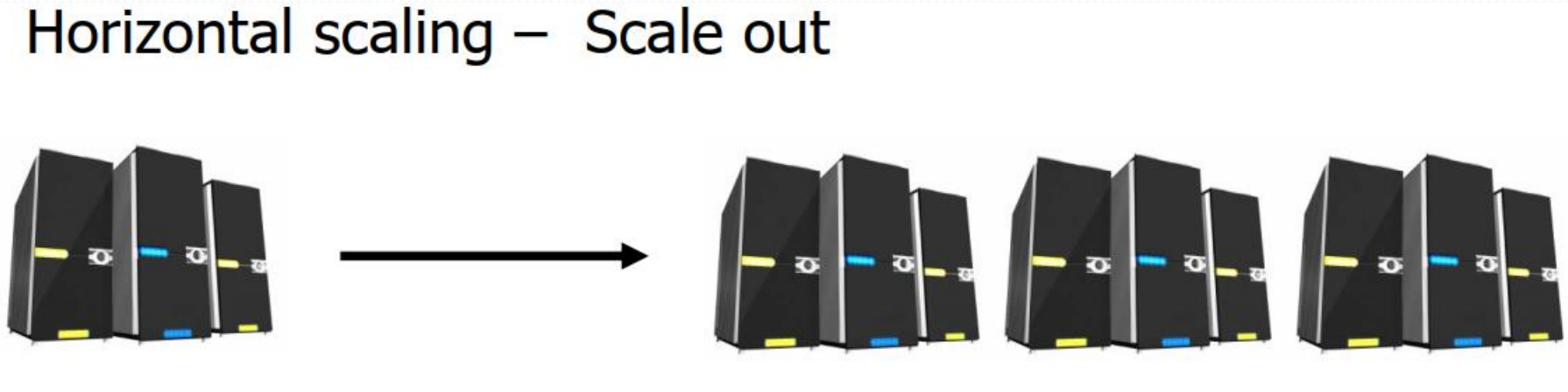
- Bối cảnh dữ liệu hiện nay: BigData
  - Dung lượng lớn
  - Tăng trưởng nhanh
  - Sự đa dạng của dữ liệu



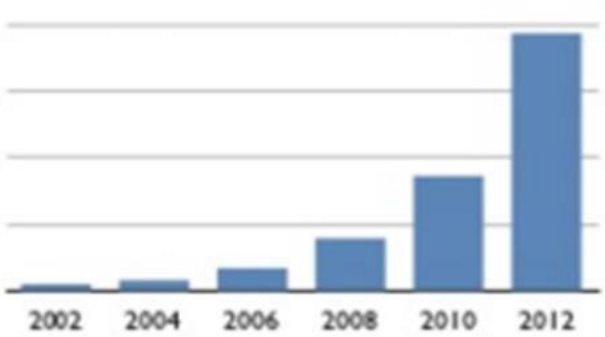
Giới thiệu

- Mở rộng hệ thống





Xu hướng hiện nay



Big data

Connectivity

P2P Knowledge

Concurrency

Diversity

Cloud-Grid

Giới thiệu

- Mô hình quan hệ:
  - Thiết kế cho mọi mục đích
  - Thỏa mãn tính chất ACID
  - Ngôn ngữ chuẩn SQL
  - Nhưng không phân tán được.

Giới thiệu

- Công nghệ xử lý, lưu trữ phát triển, hạ giá thành.
- Điện toán đám mây phát triển thực hiện ảo hóa.
- Lưu trữ phân tán xuất hiện
  - Google có hệ thống GFS (Google File System), 2003.
  - Hadoop có hệ thống HDFS (Hadoop File System), 2005.
- Xử lý song song theo mô hình MapReduce
  - Google 2004
  - Hadoop 2005

Giới thiệu

- Nhu cầu cần có những mô hình dữ liệu:
  - Khả năng lưu trữ lớn
  - Lưu dữ liệu đa dạng
  - Có khả năng mở rộng nhất là mở rộng theo chiều ngang và phân tán
  - Tính sẵn sàng cao
  - Tính nhất quán có thể yếu
- NoSQL ra đời

NoSQL là gì?

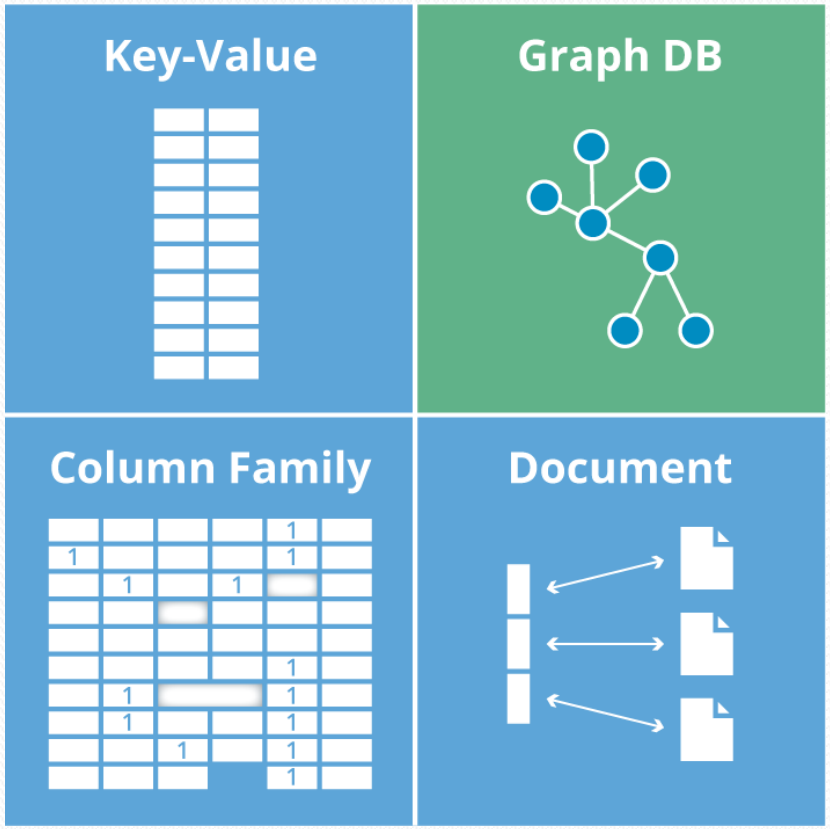
- Thuật ngữ *NoSQL* được sử dụng bởi Carlo Strozzi vào năm 1998 để đặt tên cho cơ sở dữ liệu quan hệ mã nguồn mở Strozzi NoSQL.
- Năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL trong một hội thảo về cơ sở dữ liệu nguồn mở phân tán.
- Ban đầu ngữ nghĩa của NoSQL là: distributed (phân tán) + non-relational (không ràng buộc).

NoSQL là gì?

- NoSQL là thế hệ cơ sở dữ liệu non-relational (không ràng buộc), distributed (phân tán), open source, horizontal scalable (khả năng mở rộng theo chiều ngang) có thể lưu trữ, xử lý từ một lượng rất nhỏ cho tới hàng petabytes dữ liệu trong hệ thống có độ chịu tải, lỗi cao với những đòi hỏi về tài nguyên phần cứng thấp.

NoSQL là gì?

- Website về NoSQL: http://nosql-database.org/
- Hiện có hơn 225 HQTCSDL NoSQL.
- Phân loại:
  - **Column Families**
  - **Document**
  - **Key Value**
  - **Graph**



Đặc trưng của NoSQL

- **Non-relational**
- **Schema-free**
- **Open Source**
- **Simple API**
- **Distributed**
- **Eventual consistency**

Non-relational

- Cách tiếp cận không theo mô hình quan hệ
- Không làm việc với các bảng mà các cột cố định
- Làm việc theo kiểu khép kín và phân cấp
- Không cần ánh xạ đối tượng – quan hệ và chuẩn hóa dữ liệu
- Không có các tính năng phức tạp như: hoạch định truy vấn, tính toàn vẹn tham chiếu, kết nối, ACID

Schema-free

- CSDL NoSQL không có lược đồ hoặc lược đồ mềm dẻo
- Không cần định nghĩa bất kỳ loại lược đồ nào của dữ liệu
- Cho phép cấu trúc không đồng nhất của dữ liệu trong cùng một miền

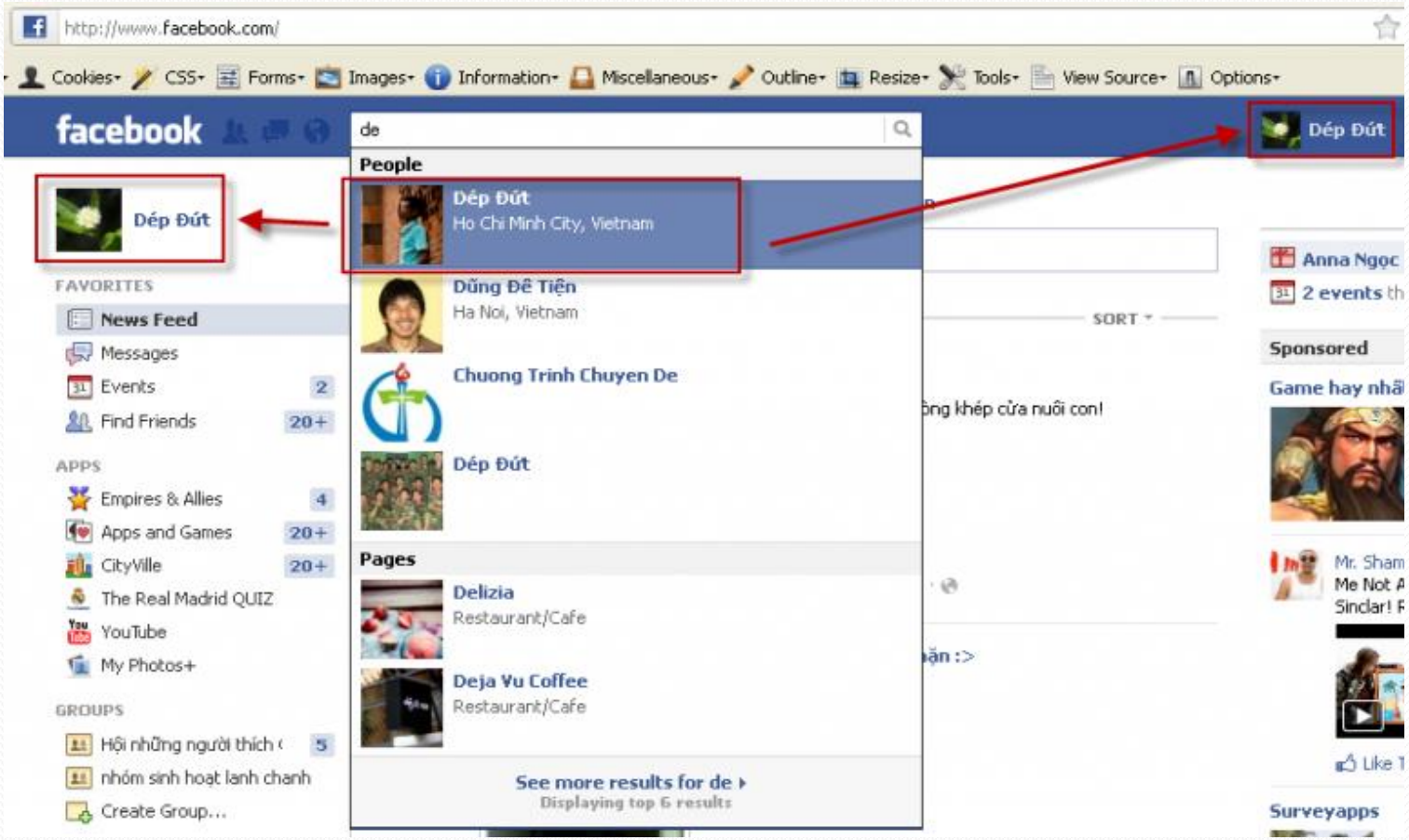
Simple API

- Cung cấp giao tiếp đơn giản để thao tác với dữ liệu và truy vấn
- API cho phép thao tác với dữ liệu mức thấp
- Không sử dụng ngôn ngữ truy vấn chuẩn
- Giao thức dựa trên văn bản thường dùng HTTP REST với JSON
- CSDL hỗ trợ web và chạy như dịch vụ trên internet

Distributed

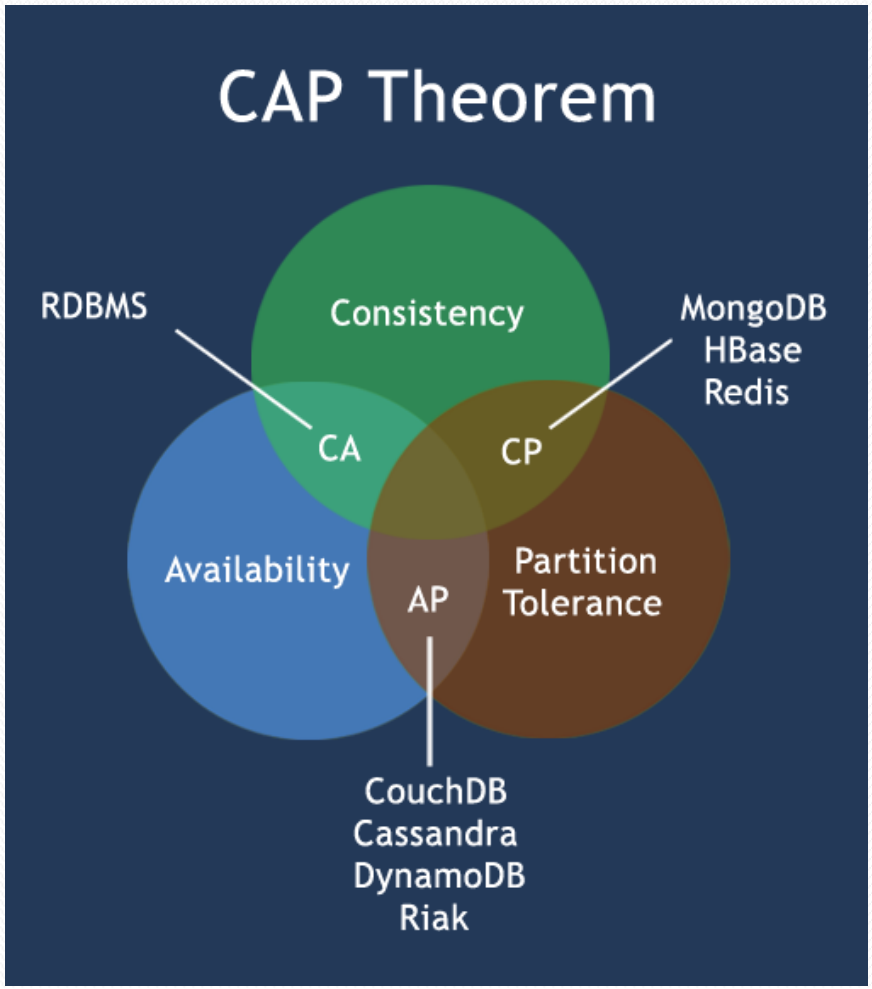
- Một số CSDL NoSQL hỗ trợ lưu trữ phân tán
- Hỗ trợ tự động mở rộng và khả năng dự phòng
- Tính chất ACID hy sinh cho khả năng mở rộng và thông lượng truy cập
- Không đồng bộ giữa các dữ liệu phân tán
- Chỉ hỗ trợ nhất quán cuối

Eventual consistency



Một số khái niệm

- Định lý CAP (Eric Brewer, 2000)
- Trong một hệ thống lưu trữ phân tán không thể đồng thời đảm bảo nhiều hơn hai trong ba tính chất sau:
  - Consistency (nhất quán)
  - Availability (sẵn sàng)
  - Partition Tolerance (thứ lỗi phân vùng)



Một số khái niệm

- **Consistency**: Mỗi lần đọc dữ liệu, sẽ nhận được nội dung mới nhất hoặc lỗi.
- **Availability**: Một yêu cầu được phản hồi lại không phải lỗi sẽ luôn được dữ liệu lưu trữ mới nhất.
- **Partition Tolerance**: Hệ thống tiếp tục hoạt động bất chấp một lượng tùy ý các thông điệp/gói tin bị mất hoặc trì hoãn do trục trặc mạng giữa các nút.

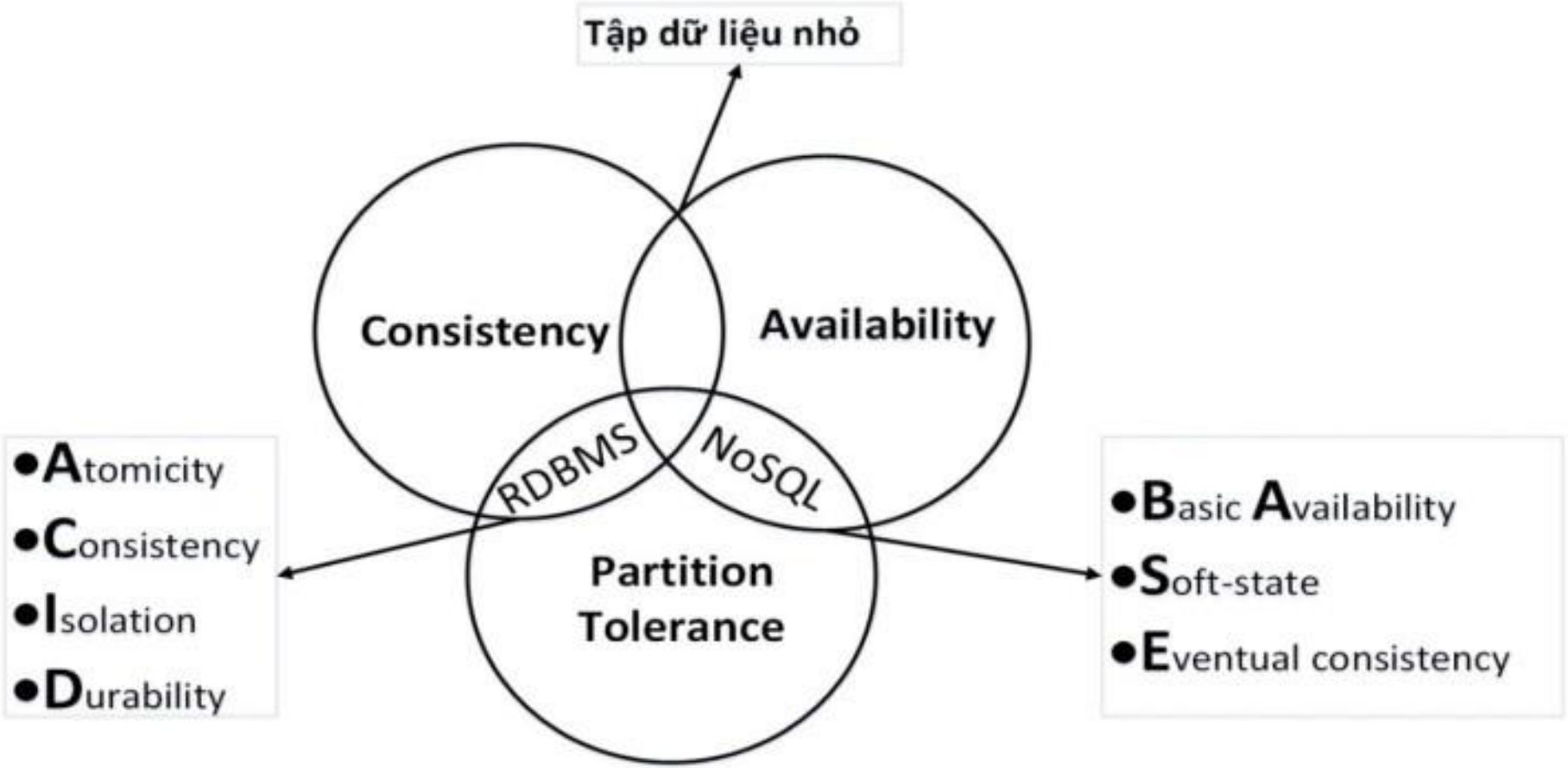
- Định lý CAP phân loại các hệ thống thành ba loại:
  - **CP (Consistent and Partition Tolerant)**
  - **CA (Consistent and Available):** các CSDL độc lập không phân tán.
  - **AP (Avaiable and Partition Tolerant)**

Nhất quán cuối

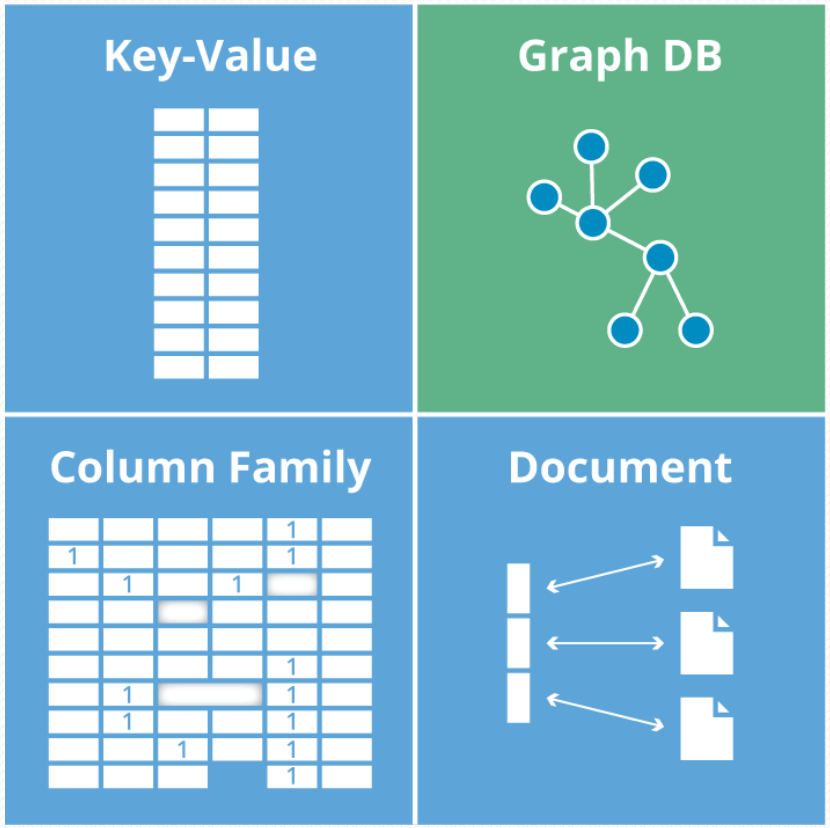
- Nhất quán cuối:
  - Bản sao của dữ liệu được chép trên nhiều máy để đảm bảo tính sẵn sàng và khả năng mở rộng.
  - Việc sửa dữ liệu trên một máy được truyền sang các máy khác chứa các bản sao
  - Việc truyền thông không tức thời nên các bản sao không nhất quán ngay lập tức
  - Các bản sao có thể trả về các giá trị khác nhau cho cùng một đơn vị dữ liệu
  - Ưu tiên cho việc truy cập nhanh
  - Những thay đổi cuối cùng được truyền tới các bản sao

BASE

- BASE: **B**asically **A**vailable, **S**oft state, **E**ventual consistency
- **B**asically **A**vailable: hệ thống đảm bảo được tính sẵn sàng ở hầu hết các trường hợp
- **S**oft state: trạng thái của hệ thống có thể thay đổi theo thời gian
- **E**ventual consistency: hệ thống sẽ nhất quán theo thời gian


Tập dữ liệu nhỏ
Consistency
Availability
RDBMS
NoSQL
Partition Tolerance
•Atomicity
•Consistency
•Isolation
•Durability
•Basic Availability
•Soft-state
•Eventual consistency

Các mô hình CSDL NoSQL



Key-Value

- Đây là mô hình đơn giản nhất
- Dữ liệu lưu theo từng cặp <key, value>
- Thích hợp lưu những dữ liệu đơn giản như dữ liệu từ thiết bị IoT, dữ liệu tạm thời trong cache,...

Session data

```
sessionid=A08154711
userlogin="xyz"
date_of_expiry=2015/12/31
```

} id

User profiles

```
{
   "id": "4",
   "name": "Mark Zuckerberg",
   "first_name": "Mark",
   "last_name": "Zuckerberg",
   "link": "http://www.facebook.com/zuck",
   "username": "zuck",
   "gender": "male",
   "locale": "en_US"
}
```

User profile id

Sensor data (IOT)

key value

timestamp	x	y	z	temperature
01.01.2014	350	120	78	-10°
01.01.2014	350	120	95	-9
01.01.2014	350	100	78	-10°
02.01.2014	350	120	78	-5°
02.01.2014	350	120	95	-8°

Một số HQTCSDL Key-Value

riak

Memcached

redis

Document

- Dùng để lưu dữ liệu tổng hợp
- Không có lược đồ cố định
- Cấu trúc nằm ở bên trong
- Thường dùng định dạng JSON, BSON,...
- Thường dùng:
  - Lưu những nội dung đơn giản như blog, nhật ký sự kiện,
  - Ứng dụng thương mại điện tử: lưu các sản phẩm và đơn hàng.

Ví dụ

```
{
    "firstName": “Paul",
    "lastName": “Adam",
    "age": 45,
    "address":
    {
        "streetAddress": "22 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
    }
}
```

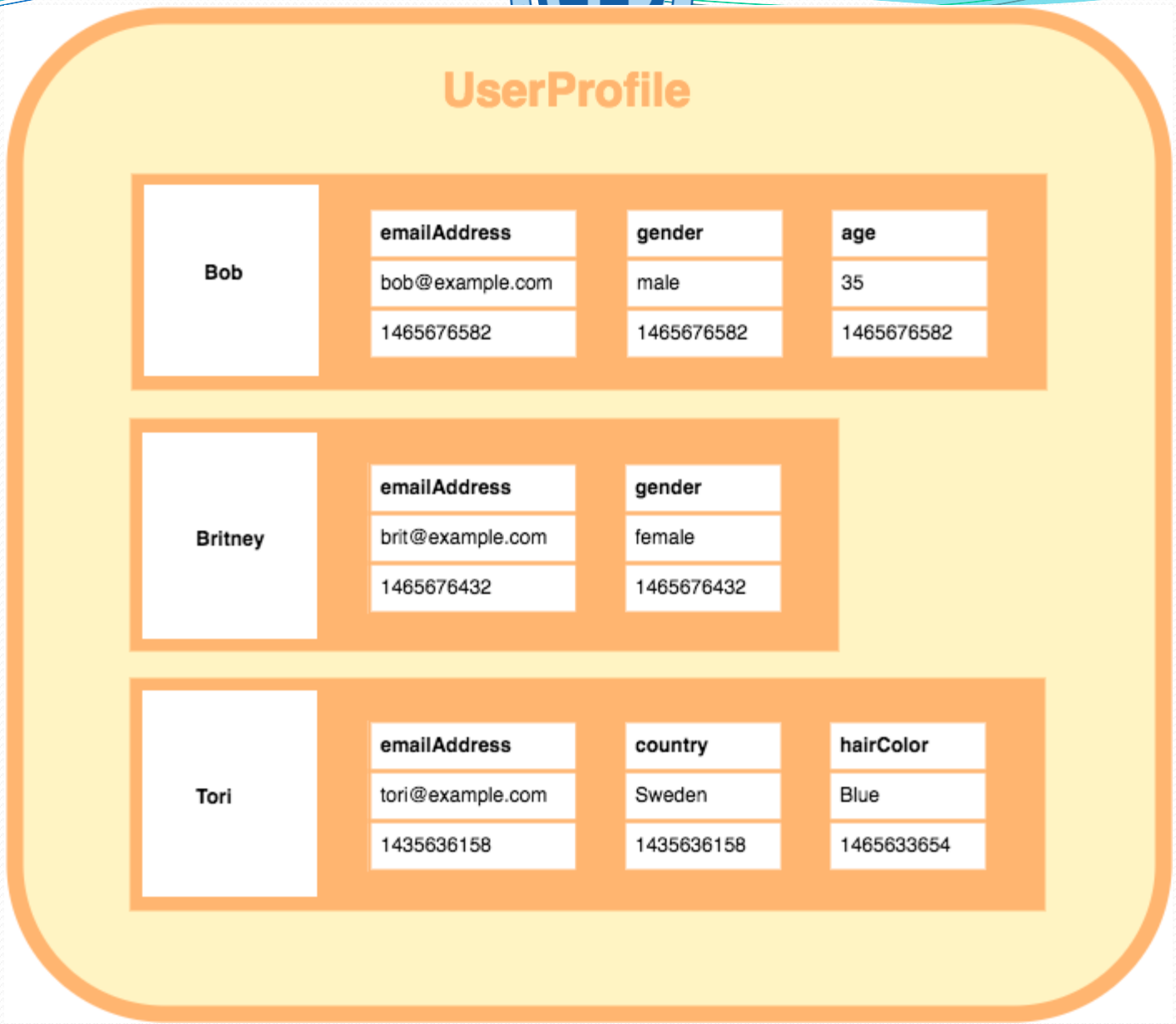

mongoDB


OrientDB®


Cloudant

Column Family

- Dữ liệu tổ chức gần như dạng bảng
- Mỗi dòng có thể có các cột khác nhau
- Mỗi dòng có một khóa và một tập các dữ liệu tương ứng với các cột
- Thường dùng để lưu các dữ liệu có cấu trúc phức tạp, không đồng nhất.

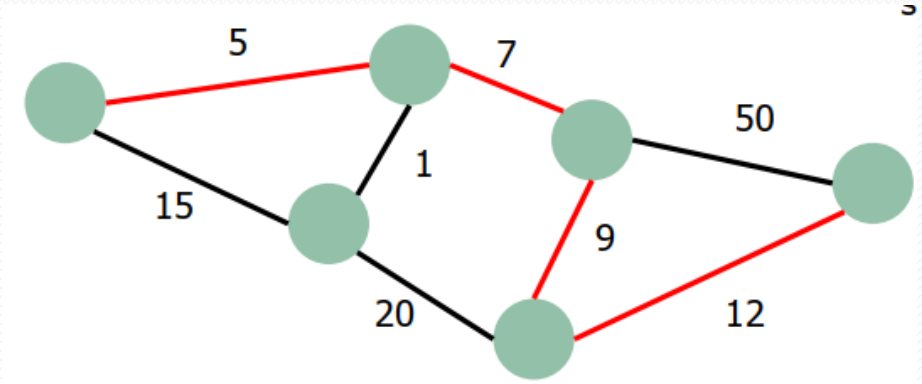

UserProfile
Bob
emailAddress
bob@example.com
1465676582
gender
male
1465676582
age
35
1465676582
Britney
emailAddress
brit@example.com
1465676432
gender
female
1465676432
Tori
emailAddress
tori@example.com
1435636158
country
Sweden
1435636158
hairColor
Blue
1465633654


APACHE
HBASE


HYPERTABLE INC

Graph

- Sử dụng mô hình đồ thị
- Dữ liệu gồm các đối tượng và các mối quan hệ
- Thường dùng để lưu những dữ liệu phức tạp có nhiều mối quan hệ




neo4j


OrientDB

Ưu, nhược điểm NoSQL

- **Ưu điểm**
  - **Open source**
  - **Khả năng mở rộng linh hoạt**
  - **Các CSDL NoSQL linh hoạt cho những dự án khác nhau**
  - **NoSQL được các hãng lớn sử dụng:** Amazon, BBC, Facebook, Google,…
  - **NoSQL phù hợp với công nghệ đám mây**

Ưu, nhược điểm NoSQL

- **Nhược điểm**
  - Tính nhất quán chưa được đảm bảo tức thời
  - Chưa hỗ trợ tốt cho doanh nghiệp
  - Những vấn đề về tính tương thích

Tổng kết

- Cơ sở dữ liệu NoSQL là xu hướng dữ liệu phù hợp xu hướng hiện nay
- NoSQL cần phát triển nhiều hơn về kỹ thuật và chuẩn hóa
- Việc sử dụng NoSQL thực sự linh hoạt

Câu hỏi

- NoSQL có thay thế cho SQL được không?
- Tìm hiểu Facebook dùng SQL hay NoSQL?
- NoSQL đã đáp ứng được nhu cầu lưu trữ dữ liệu của con người chưa?