



CƠ SỞ DỮ LIỆU LƯU TRỮ VÀ QUẢN LÝ DỮ LIỆU LỚN

Giảng viên: Nguyễn Tu Trung, Trần Mạnh Tuấn
BM HTTT, Khoa CNTT, Trường ĐH Thủy Lợi

Hà Nội, 2019

Nội dung

- ❖ CSDL SQL
- ❖ CSDL NoSQL
- ❖ So sánh giữa CSDL SQL và NoSQL
- ❖ Phân loại CSDL NoSQL
- ❖ CSDL MongoDB
- ❖ CSDL Cassandra

CSDL SQL

- ❖ Database:
 - ❖ Là cơ sở dữ liệu: gồm các bảng, hàng, cột
- ❖ MySQL, Microsoft SQL server, Oracle,...:
 - ❖ Là hệ quản trị cơ sở dữ liệu (DBMS)
- ❖ SQL:
 - ❖ Viết tắt của Structure Query Language (Ngôn ngữ truy vấn cấu trúc)
 - ❖ Truy vấn trên nền một RDBMS (Hệ quản trị CSDL quan hệ)
- ❖ Trong RDBMS, dữ liệu được lưu vào nhiều bảng, mỗi bảng sẽ có nhiều cột, nhiều hàng
- ❖ Ta sử dụng SQL để truy vấn như sau:
 - ❖ `SELECT Name, Age FROM Students WHERE ...`

CSDL SQL

- ❖ Lý do RDBMS được sử dụng phổ biến, trong hầu hết các ứng dụng:
 - ❖ Đảm bảo tính (Atomicity-Nguyên tử, Consistency-Nhất quán, Isolation-Độc lập, Durability-Bền vững) của transaction
 - ❖ Với database chuẩn 3, dữ liệu được đảm bảo tính đồng nhất và toàn vẹn (consistency)
 - ❖ Rất nhiều ngôn ngữ làm việc với RDBMS: Java, C#, PHP
 - ❖ Số lượng lập trình viên biết và dùng SQL rất rất nhiều

CSDL SQL

❖ Hạn chế:

- ❖ Việc mapping giữa các bảng trong database với các object trong code khá rắc rối và phức tạp
- ❖ Performance sẽ bị chậm khi phải join nhiều bảng để lấy dữ liệu => lý do sử dụng “giảm chuẩn” để tăng hiệu suất cho RDBMS
- ❖ Việc thay đổi cấu trúc dữ liệu (Thêm/xóa bảng hoặc thêm/xóa các trường) rất mệt mỏi => Có thể vi phạm tính toàn vẹn, tính nhất quán CSDL, thay đổi trên code => Thiếu linh hoạt, cứng nhắc
- ❖ Yêu cầu cấu trúc CSDL rõ ràng, không làm việc được với dữ liệu không có cấu trúc (unstructure)
- ❖ RDBMS được thiết kế để chạy trên một máy chủ, nếu muốn mở rộng, nó khó chạy trên nhiều máy (clustering)

CSDL NoSQL

❖ NoSQL:

- ❖ Nghĩa là Not Only SQL (Không chỉ SQL) => ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu
- ❖ Còn có nghĩa là Non-Relational (NoRel) - không ràng buộc (hiện nay ít phổ dụng hơn)
- ❖ Được giới thiệu lần đầu vào năm 1998
- ❖ Ban đầu được sử dụng bởi các “ông lớn” trên Internet như Facebook, Google, Amazon và những người yêu cầu hệ thống quản lý cơ sở dữ liệu: có thể viết và đọc dữ liệu bất cứ nơi nào trên thế giới, trong khi mở rộng quy mô và cung cấp hiệu suất trên toàn bộ dữ liệu lớn và hàng triệu người sử dụng

CSDL NoSQL

- ❖ NoSQL:
 - ❖ Nghĩa là Not Only SQL (Không chỉ SQL) => ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu
 - ❖ Còn có nghĩa là Non-Relational (NoRel) - không ràng buộc (hiện nay ít phổ dụng hơn)
- ❖ Được giới thiệu lần đầu vào năm 1998
- ❖ Ban đầu được sử dụng bởi các “ông lớn” trên Internet như Facebook, Google, Amazon và những người yêu cầu hệ thống quản lý cơ sở dữ liệu: có thể viết và đọc dữ liệu bất cứ nơi nào trên thế giới, trong khi mở rộng quy mô và cung cấp hiệu suất trên toàn bộ dữ liệu lớn và hàng triệu người sử dụng
- ❖ Dữ liệu được lưu dưới dạng object, mặc dù bị trùng lặp nhưng truy vấn rất nhanh và đơn giản

CSDL NoSQL

- ❖ NoSQL đánh dấu bước phát triển của thể hệ database mới: distributed (phân tán) và non-relational (không ràng buộc)
- ❖ CSDL NoSQL ra đời giải quyết được những khuyết điểm của CSDL SQL
 - ❖ Dữ liệu trong NoSQL DB được lưu dưới dạng document, object => Truy vấn dễ dàng và nhanh hơn RDBMS nhiều
 - ❖ NoSQL có thể làm việc tốt với dữ liệu dạng không có cấu trúc
 - ❖ Việc đổi cấu trúc dữ liệu (Thêm, xóa trường hoặc bảng) rất dễ dàng và nhanh gọn trong NoSQL
 - ❖ Không đặt nặng tính ACID của transactions và tính nhất quán của dữ liệu => NoSQL DB có thể mở rộng, chạy trên nhiều máy một cách dễ dàng

Một số thuật ngữ liên quan

- ❖ Relational (ràng buộc):
 - ❖ Quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ sử dụng mô hình khóa chính- phụ để ràng buộc dữ liệu
 - ❖ Thể hiện tính nhất quán dữ liệu từ các bảng khác nhau
- ❖ Non-relational (Không ràng buộc):
 - ❖ Không sử dụng các ràng buộc dữ liệu cho nhất quán dữ liệu trong CSDL NoSQL
- ❖ High Availability (Tính sẵn sàng):
 - ❖ Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống
- ❖ High Scalability (Khả năng mở rộng):
 - ❖ Gần như không có một giới hạn cho dữ liệu và người dùng trên hệ thống

Một số thuật ngữ liên quan

- ❖ Eventual consistency (Nhất quán cuối):
 - ❖ Dữ liệu không cần phải đảm bảo ngay tức khắc sau mỗi phép write
 - ❖ 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 => cuối cùng (eventually) dữ liệu trên hệ thống sẽ trở lại trạng thái nhất quán
- ❖ Vertical scalable (Khả năng mở rộng chiều dọc):
 - ❖ Là phương pháp tăng cường khả năng lưu trữ và xử lý (Khi dữ liệu lớn về lượng) bằng việc cải tiến phần mềm và cải thiện phần cứng trên một máy tính đơn lẻ
 - ❖ Ví dụ: tăng cường CPUs, cải thiện đĩa cứng, bộ nhớ trong một máy tính

Một số thuật ngữ liên quan

- ❖ Horizontal scalable (Khả năng mở rộng chiều ngang):
 - ❖ Là phương pháp tăng cường khả năng lưu trữ và xử lý (Khi dữ liệu lớn về lượng) mà dùng nhiều máy tính phân tán
 - ❖ Phân tán và dữ liệu được hỗ trợ bởi phần mềm tức cơ sở dữ liệu
- ❖ Distributed Data (Phân tán dữ liệu):
 - ❖ Mô hình lưu trữ phân tán các file hoặc dữ liệu ra nhiều máy tính khác nhau trong mạng LAN hoặc Internet dưới sự kiểm soát của phần mềm
- ❖ Deployment Flexibility (Triển khai linh hoạt):
 - ❖ Việc bổ sung thêm/loại bỏ các node, hệ thống sẽ tự động nhận biết để lưu trữ mà không cần phải can thiệp bằng tay
 - ❖ Hệ thống không đòi hỏi cấu hình phần cứng mạnh, đồng nhất
- ❖ Durability (Lưu trữ tốt):
 - ❖ Dữ liệu có thể tồn tại trong bộ nhớ máy tính nhưng đồng thời cũng được lưu trữ lại đĩa cứng

So sánh giữa CSDL SQL và NoSQL

- ❖ Hiệu suất:
 - ❖ SQL kém NoSQL phải kiểm tra và xử lý quan hệ, ràng buộc
 - ❖ NoSQL rất tốt vì bỏ qua quan hệ, ràng buộc
- ❖ Khả năng mở rộng theo chiều ngang
 - ❖ SQL: Hạn chế về số lượng, khả năng mở rộng các node
 - ❖ NoSQL: Hỗ trợ một lượng rất lớn các node
- ❖ Yêu cầu phần cứng:
 - ❖ SQL: Cấu hình cao hoặc đặc biệt
 - ❖ NoSQL: Không đòi hỏi cấu hình cao
- ❖ Tính nhất quán dữ liệu
 - ❖ SQL: Đảm bảo tính nhất quán và ràng buộc toàn vẹn dữ liệu
 - ❖ NoSQL: Tính nhất quán và ràng buộc toàn vẹn yếu và chủ yếu được thực thi nhiều trên code

So sánh giữa CSDL SQL và NoSQL

- ❖ Tốc độ đọc ghi
 - ❖ SQL: Chậm hơn NoSQL vì: (1) phải đảm bảo tính ràng buộc giữa các bảng, (2) nếu nhiều server thì phải đảm bảo tính nhất quán về dữ liệu giữa các server
 - ❖ NoSQL: Nhanh hơn SQL vì: (1) bỏ đi các cơ chế ràng buộc, toàn vẹn dữ liệu; (2) dữ liệu được xử lý trong bộ nhớ rồi mới ghi từ từ xuống đĩa (đảm bảo tính nhất quán cuối)
- ❖ Thay đổi số node trong hệ thống
 - ❖ SQL: thường phải shutdown cả hệ thống, việc thay đổi số node phức tạp
 - ❖ NoSQL: Không cần phải shutdown cả hệ thống, việc thay đổi số node đơn giản, ảnh hưởng rất ít đến hệ thống
- ❖ Dữ liệu:
 - ❖ SQL: Có cấu trúc; NoSQL: Cả cấu trúc, bán cấu trúc và phi cấu trúc

So sánh giữa CSDL SQL và NoSQL

- ❖ Truy vấn và báo cáo
 - ❖ SQL: Dễ dàng sử dụng ngôn ngữ SQL query để truy vấn trực tiếp dữ liệu từ database hoặc dùng công cụ hỗ trợ để lấy báo cáo
 - ❖ NoSQL: Việc lấy báo cáo dữ liệu trực tiếp từ NoSQL chưa được hỗ trợ tốt, thực hiện chủ yếu thông qua giao diện ứng dụng
- ❖ Ứng dụng:
 - ❖ SQL: Phù hợp với hệ thống có quan hệ chặt chẽ và nhất quán về dữ liệu: Tài chính, ngân hàng, chứng khoán...
 - ❖ NoSQL: Phù hợp với hệ thống lưu trữ thông tin lớn, không quá quan trọng về tính nhất quán dữ liệu trong khoảng thời gian nhất định: Báo chí, mạng xã hội, diễn đàn...

Phân loại CSDL NoSQL

- ❖ Khóa – giá trị (Key – value)
- ❖ Tài liệu (Document-based)
- ❖ Cột (Column-Family)
- ❖ Đồ thị (Graph-based)

CSDL Key-Value

- ❖ Hệ quản trị CSDL Key-Value lưu trữ dữ liệu dưới dạng Key (là một chuỗi duy nhất) liên kết với Value có thể ở dạng chuỗi văn bản đơn giản hoặc các tập, danh sách dữ liệu phức tạp hơn
- ❖ Ví dụ dữ liệu Key-Value:
 - ❖ “1234”, “Thanh Huy,zxthanhhuuy@gmail.com”
 - ❖ “1235”, “Trọng Nghĩa,letrongnghia@gmail.com,TP. HCM”
 - ❖ “1236”, “Thanh Nghị”
 - ❖ “1237”, “Trường Giang,Bình Thuận”
- ❖ Dữ liệu là danh sách các cặp Key-Value
- ❖ Là kiểu lưu trữ đơn giản nhất trong các loại CSDL NoSQL
 - ❖ Không cần quan tâm đến nội dung cần lưu trữ là gì
 - ❖ Cấu trúc, định dạng, quy ước dữ liệu trong Value do client đảm nhiệm

CSDL Key-Value

- ❖ Các API cung cấp cho việc truy vấn dữ liệu của các CSDL Key-value thường cũng rất đơn giản, phổ biến là:
 - ❖ `void Put(string key, byte[] data);`
 - ❖ `byte[] Get(string key);`
 - ❖ `void Remove(string key);`
- ❖ Quá trình tìm kiếm dữ liệu thực hiện thông qua key => Tốc độ truy suất dữ liệu rất nhanh
- ❖ Cách lưu trữ Key-Value phù hợp những trường hợp quản lý đối tượng (hàng hóa, sản phẩm, khách hàng) với Key là ID đối tượng và Value là toàn bộ thông tin về đối tượng
- ❖ Hạn chế: Nếu thông tin đối tượng phức tạp, nhiều thuộc tính thì client phải xử lý tiếp để tách các thuộc tính
- ❖ CSDL Key-Value tiêu biểu: Riak, Redis

CSDL Document

- ❖ CSDL Document được thiết kế để quản lý và lưu trữ dữ liệu ở dạng document
- ❖ document này được định dạng theo chuẩn: XML, JSON (Javascript Option Notation) hay BSON (Binary JSON)
- ❖ Dữ liệu bán cấu trúc:
 - ❖ Gồm danh sách các cặp Name-Value
 - ❖ Value là một bộ gồm danh sách các cặp SubName-SubValue
 - ❖ => Cấu trúc dữ liệu cây phân cấp
- ❖ Phù hợp cho việc lưu trữ và quản lý tập dữ liệu có kích thước lớn như là tài liệu văn bản, tin nhắn
- ❖ CSDL Document tiêu biểu: CouchDB (JSON), MongoDB (BSON)... đều là mã nguồn mở

CSDL Document

JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ]
}
```

XML

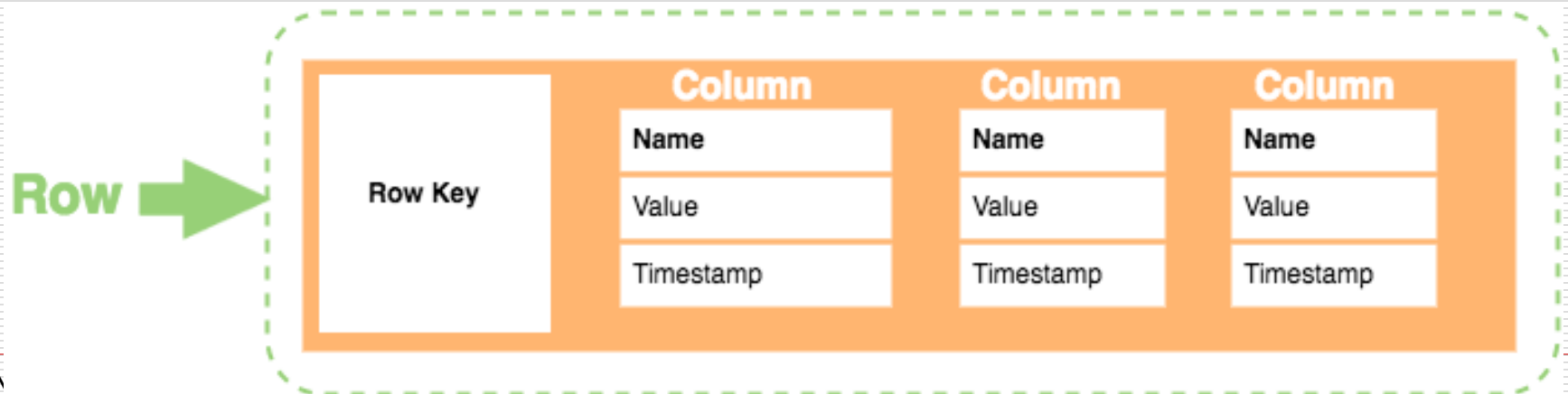
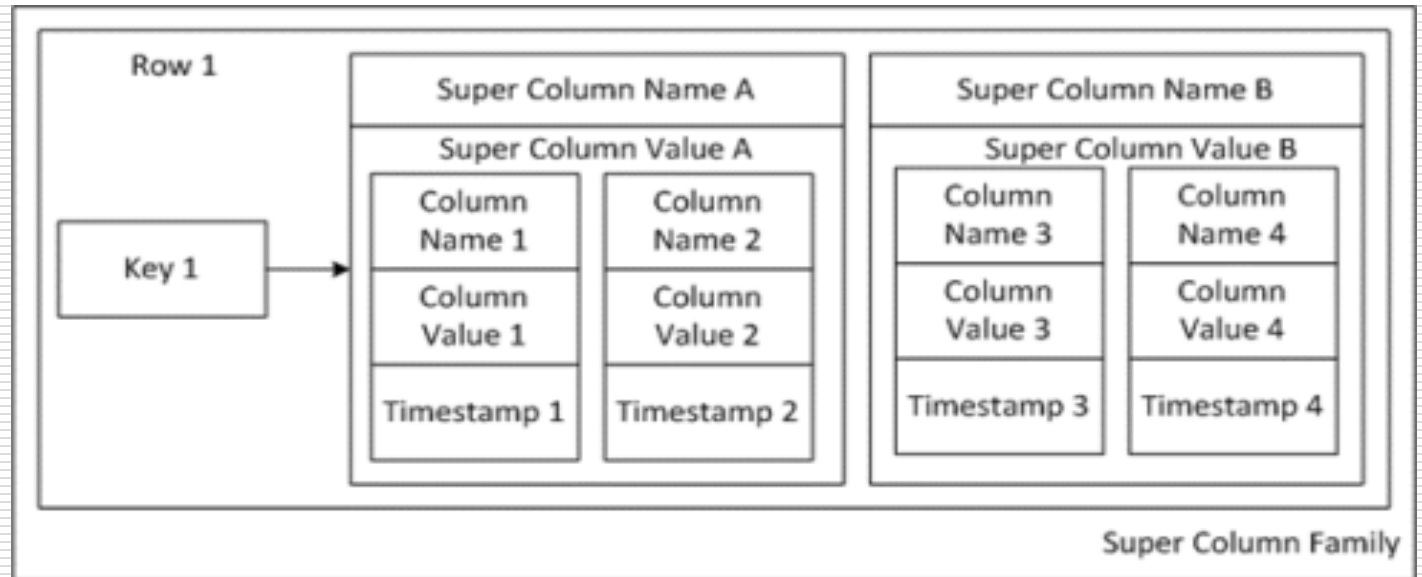
```
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumbers>
    <phoneNumber>
      <type>home</type>
      <number>212 555-1234</number>
    </phoneNumber>
    <phoneNumber>
      <type>fax</type>
      <number>646 555-4567</number>
    </phoneNumber>
  </phoneNumbers>
</person>
```

CSDL Column-Family

- ❖ Column:
 - ❖ Là bộ gồm Name, Value và TimeStamp (dấu thời gian)
 - ❖ Thường chỉ quan tâm tới name-value
- ❖ Super column:
 - ❖ Là bộ gồm Supper Name và SupperValue
 - ❖ Giá trị siêu cột gồm nhiều cột
- ❖ Column family
 - ❖ Gồm nhiều dòng
 - ❖ Mỗi dòng gồm RowKey và RowValue; nhiều column hoặc Super column
 - ❖ Tất cả dữ liệu trong một Column family sẽ được lưu trên cùng một file

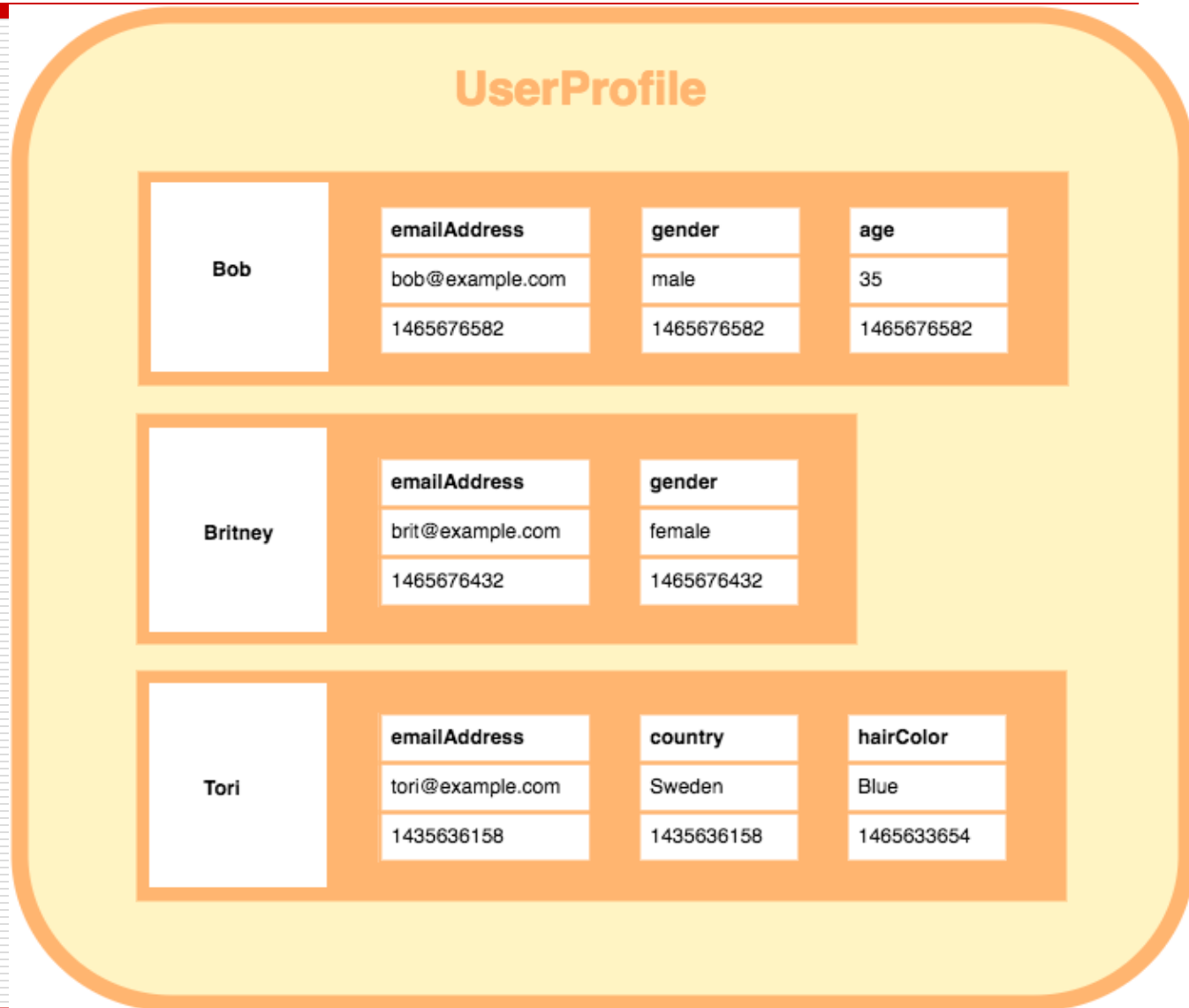
CSDL Column-Family

❖ Ví dụ: Cấu trúc Column-Family



CSDL Column-Family

- ❖ Ví dụ: lưu trữ thông tin UserProfile với Column-Family



CSDL Column-Family

- ❖ Không thể lưu trữ một lượng rất lớn dữ liệu trong cơ sở dữ liệu quan hệ
- ❖ CSDL Column Family loại bỏ các khái niệm trừu tượng, những thứ làm cho nó cứng nhắc khi chạy trên một cụm máy
- ❖ CSDL Column Family được thiết kế để chạy trên một số lượng lớn các máy, và lưu trữ một lượng dữ liệu cực lớn
- ❖ Truy vấn được các CSDL Column Family xử lý một cách dễ dàng
- ❖ CSDL cột tiêu biểu: [Hbase](#), [Cassandra](#)

CSDL Graph

- ❖ Là kiểu cơ sở dữ liệu đồ thị, gồm một tập các đối tượng gọi là đỉnh nối với nhau bởi các cạnh
- ❖ CSDL quan hệ truyền thống không phải là giải pháp hay với những bài toán cần nhiều quan hệ việc sử dụng để lưu trữ vì:
 - ❖ Việc biểu diễn quan hệ dưới dạng bảng không phải là một cách làm trực quan
 - ❖ Các phép join thường tốn rất nhiều chi phí
- ❖ 3 khái niệm cơ bản quan trọng nhất:
 - ❖ **Node**: là thực thể hay đối tượng, graph không quan tâm đến kiểu dữ liệu của thực thể
 - ❖ **Properties**: là thuộc tính đối tượng
 - ❖ **Edges**: là cạnh nối các node, là quan hệ giữa các đối tượng; tùy từng quan hệ mà cạnh có hướng hay không

CSDL Graph

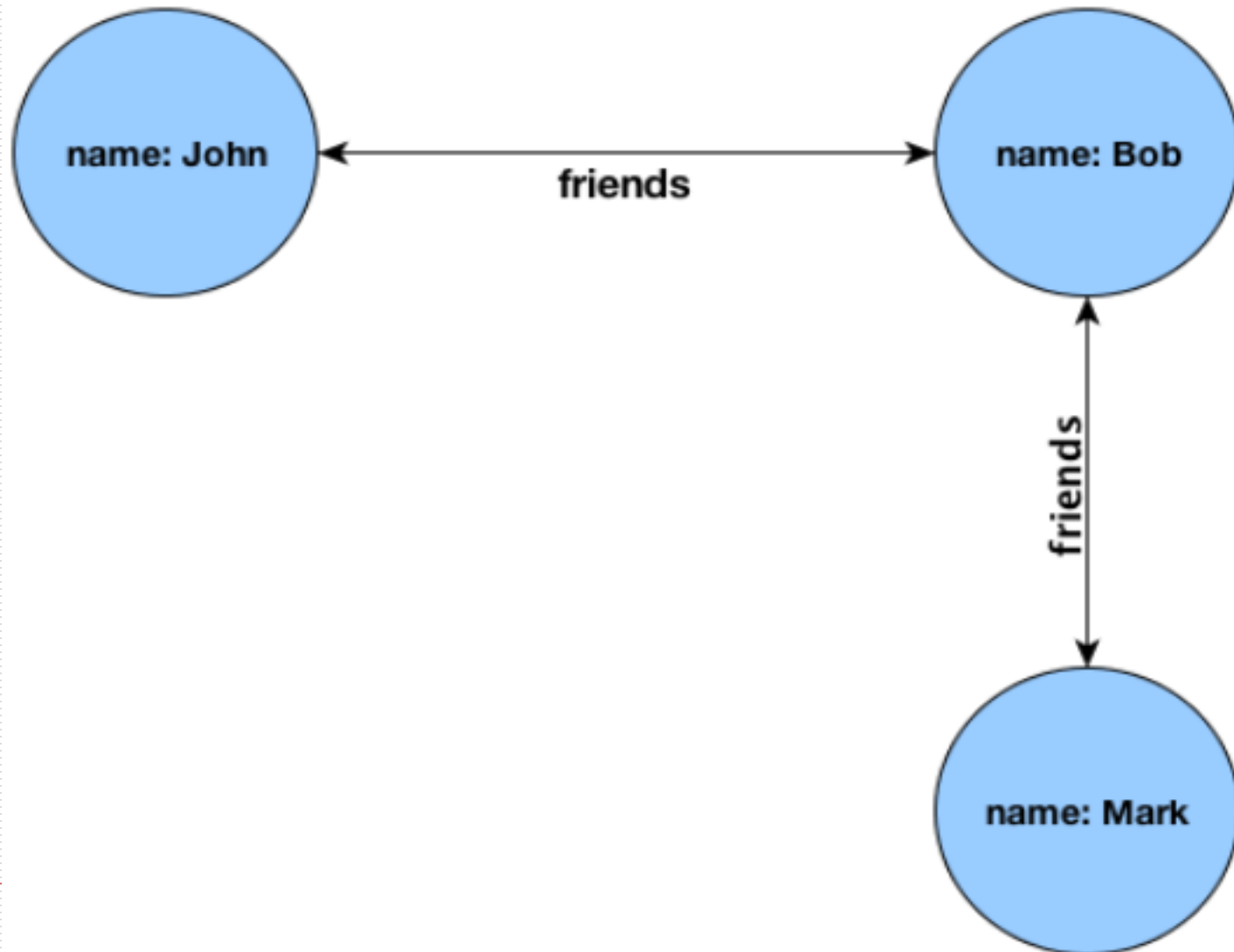
- ❖ Ví dụ: Biểu diễn quan hệ bạn bè của 2 người tên Bob và Mark
- ❖ Biểu diễn bằng các CSDL quan hệ:
 - ❖ Mỗi 1 hàng trong bảng user sẽ biểu diễn 1 người
 - ❖ Phải liệt kê quan hệ bạn bè qua bảng friends
 - ❖ Nếu ta muốn biểu diễn 1 quan hệ khác của user thì ta phải tạo bảng mới để mô tả quan hệ đó rồi liên kết với bảng user => Rất phức tạp

users	
id	name
1	John
2	Bob
3	Mark

friends		
id	user_id	user2_id
1	1	2
2	2	1
3	2	3
4	3	2

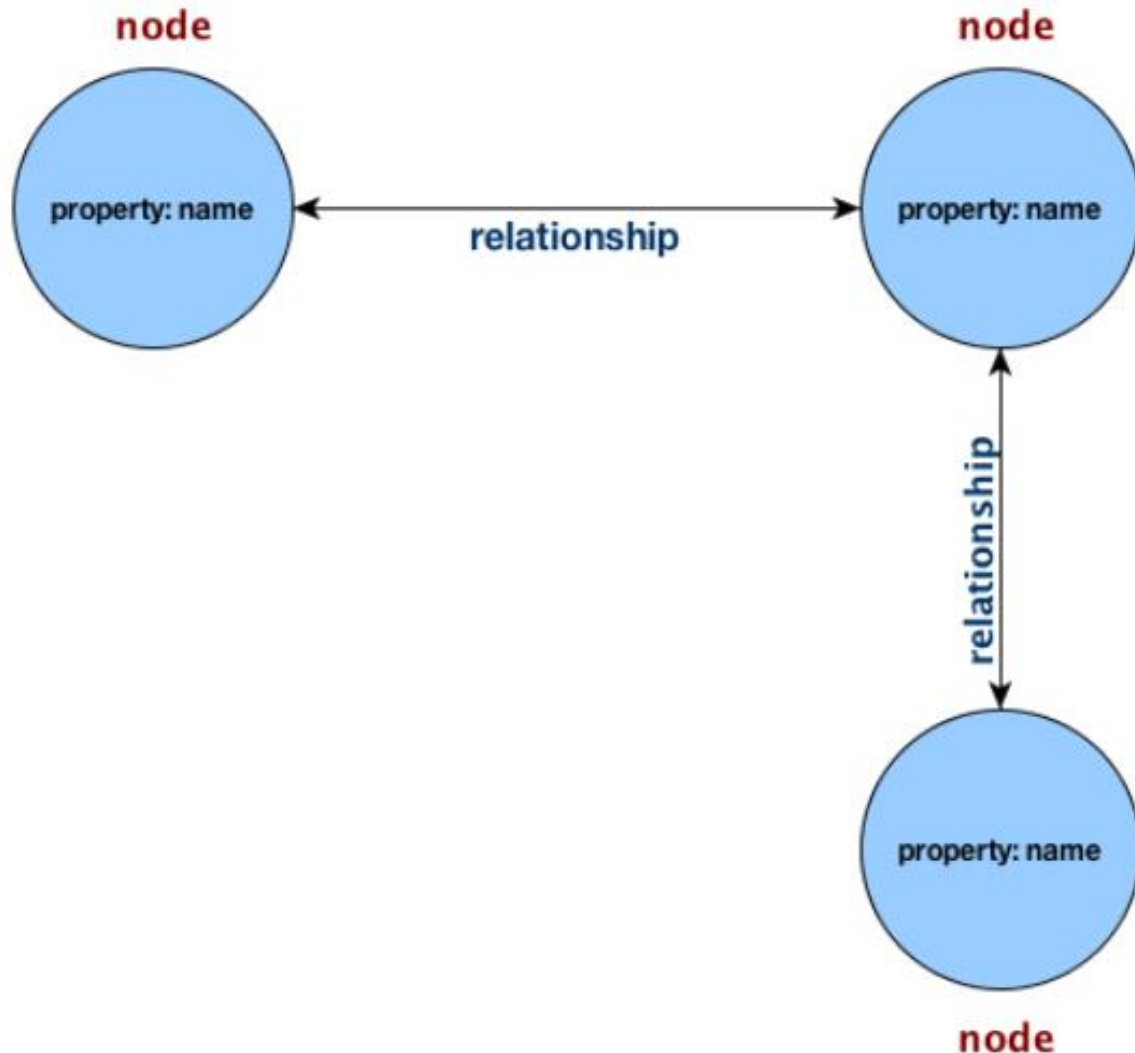
CSDL Graph

❖ Biểu diễn bằng các CSDL graph:



CSDL Graph

- ❖ Đồ thị tổng quát biểu diễn đối tượng và quan hệ:
- ❖ Một số sản phẩm tiêu biểu của graph database là: Neo4J, Sones, AllegroGraph, Core Data, DEX, FlockDB, InfoGrid, OpenLink Virtuoso...



CSDL MongoDB

- ❖ Giới thiệu về MongoDB
- ❖ Tổng quan về MongoDB
- ❖ Lợi thế của MongoDB
- ❖ Cài đặt MongoDB
- ❖ Cấu hình MongoDB

Giới thiệu về MongoDB

- ❖ Là cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng
- ❖ Được viết bằng C++
- ❖ Là cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document
- ❖ Không có ràng buộc Relationship như các hệ quản trị cơ sở dữ liệu khác nên việc truy xuất rất nhanh
- ❖ Cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng

Tổng quan về MongoDB

❖ Khái niệm Database

- ❖ Là một nơi chứa vật lý cho các Collection
- ❖ Mỗi MongoDB Server có thể có nhiều cơ sở dữ liệu

❖ Khái niệm Collection

- ❖ Là một nhóm các Document trong MongoDB
- ❖ Các Collection Mỗi collection có thể chứa nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định

❖ Khái niệm Document

- ❖ Là tập hợp các cặp key-value, có cấu trúc tương tự như kiểu dữ liệu JSON

Tổng quan về MongoDB

- ❖ Ví dụ: cấu trúc Document của một Blog site với các cặp key-value phân biệt bởi dấu phẩy
- ❖ **_id** là một số thập lục phân 12 byte để đảm bảo tính duy nhất của mỗi Document
- ❖ **_id** có thể được cung cấp bởi người dùng trong khi chèn vào Document
- ❖ Nếu người dùng không cung cấp, MongoDB sẽ cung cấp một id duy nhất cho mỗi Document

```
{
  _id: ObjectId(7df78ad8902c)
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user: 'user1',
      message: 'My first comment',
      dateCreated: new Date(2011,1,20,2,15),
      like: 0
    },
    {
      user: 'user2',
      message: 'My second comments',
      dateCreated: new Date(2011,1,25,7,45),
      like: 5
    }
  ]
}
```

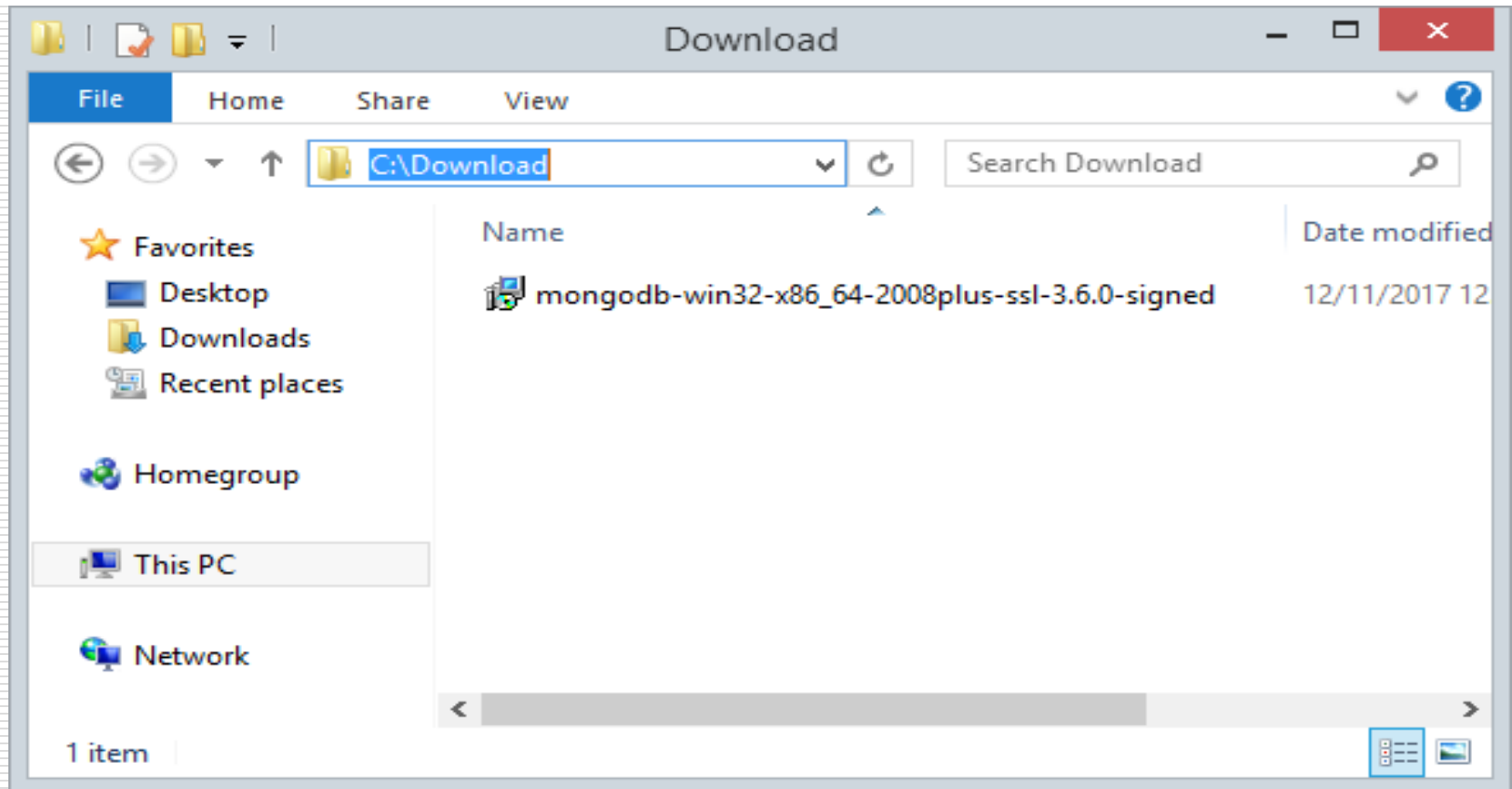
Lợi thế của MongoDB

- ❖ Document trong cùng một Collection không cần thiết phải có cùng một tập hợp các trường hoặc cấu trúc giống nhau
- ❖ Số trường, nội dung, kích thước của Document này có thể khác với Document khác
- ❖ Cấu trúc một đối tượng rõ ràng
- ❖ Không có các join phức tạp
- ❖ Khả năng truy vấn linh hoạt: MongoDB hỗ trợ truy vấn động bởi ngôn ngữ truy vấn mạnh mẽ như SQL
- ❖ Dễ dàng mở rộng, thay đổi cấu trúc vì các trường không fix

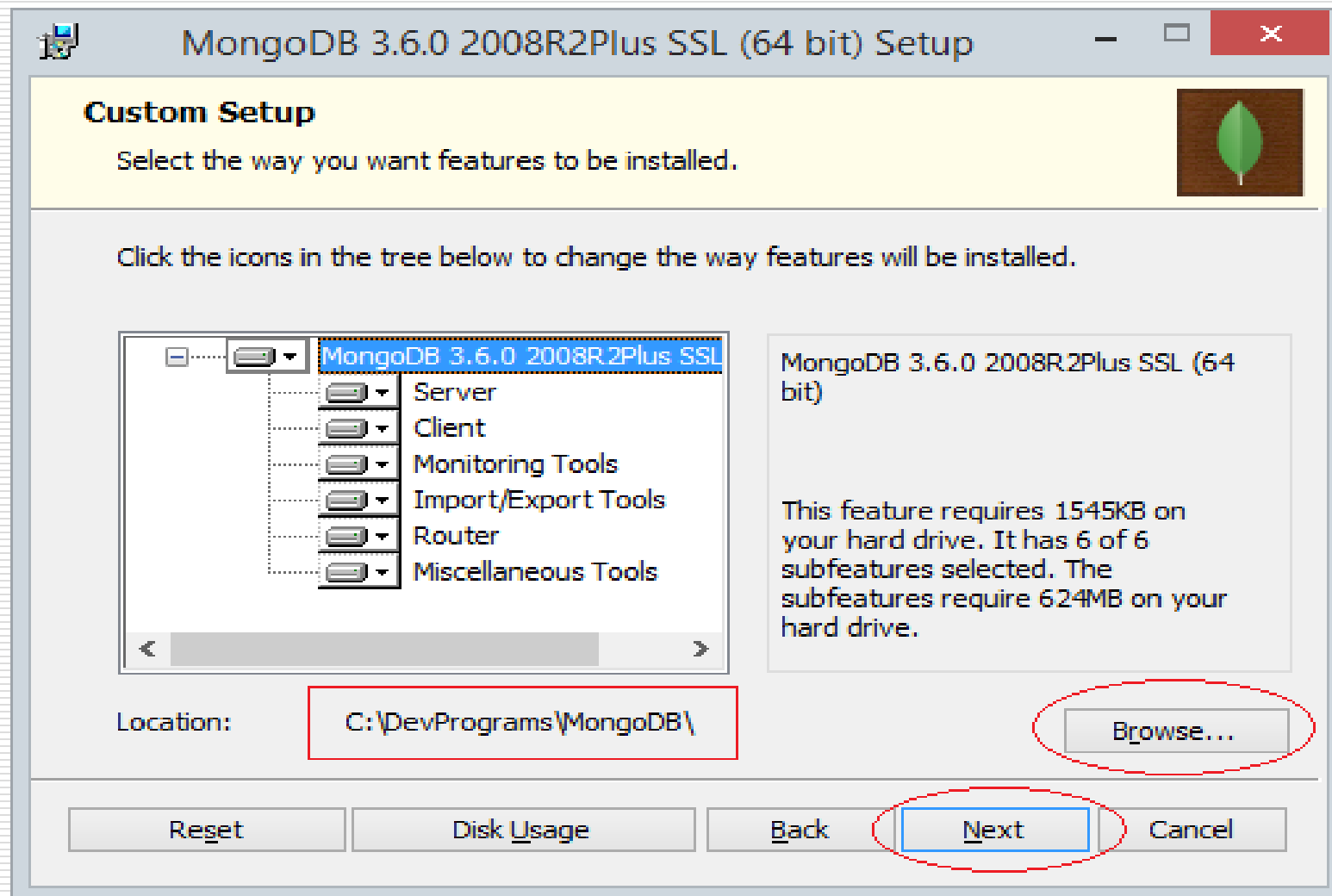
Cài đặt MongoDB

❖ Tải bộ cài đặt từ trang:

<https://www.mongodb.com/download-center#community>

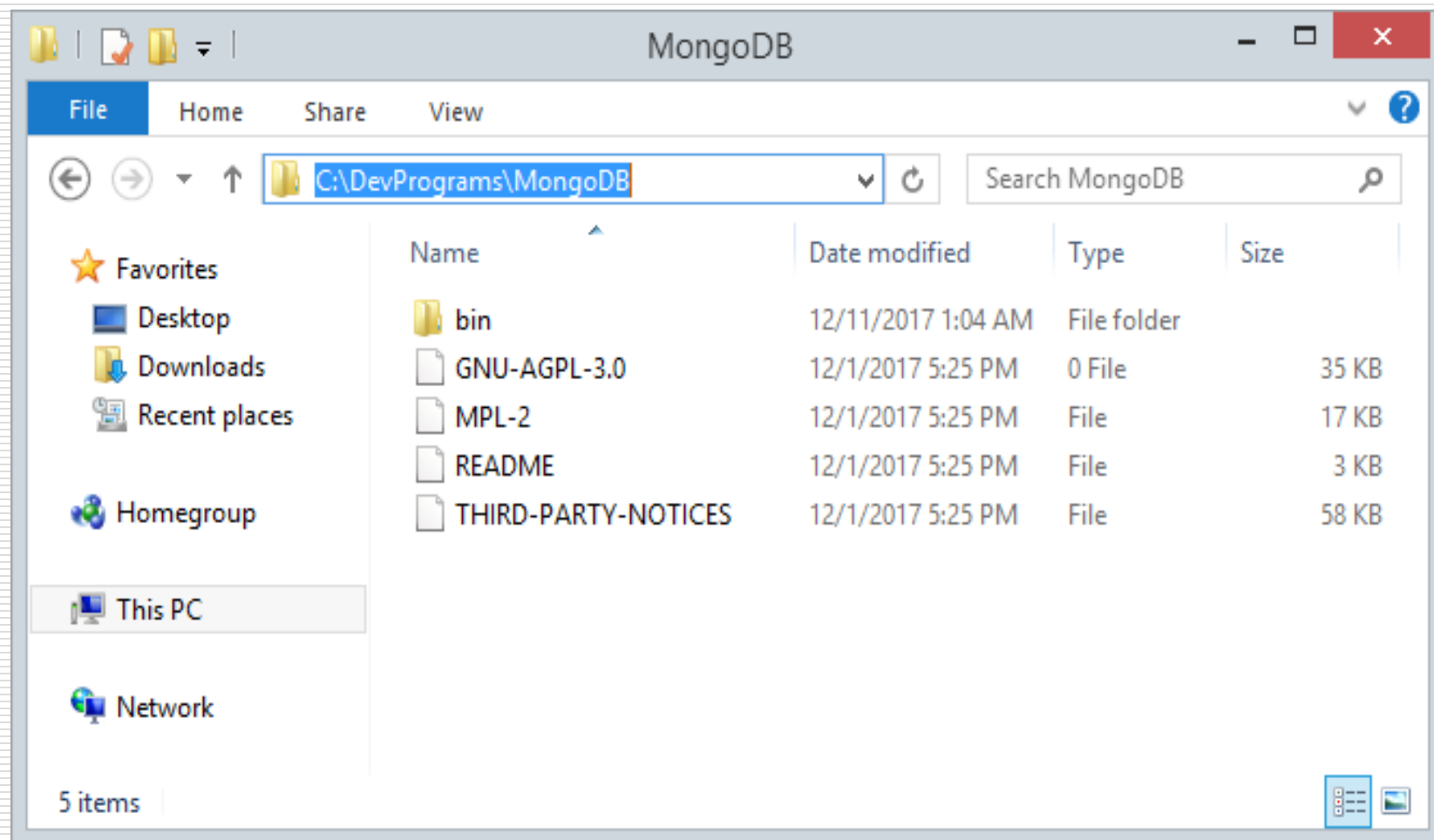


Cài đặt MongoDB



Cài đặt MongoDB

❖ Thư mục MongoDB sau khi cài đặt



Cấu hình MongoDB

- ❖ Tạo thư mục “**MongoStore**” chứa dữ liệu của Database, thư mục chứa file log, file cấu hình,... => Không nên để trong ổ C, có thể để ổ D
- ❖ Trong thư mục “**MongoStore**”
 - ❖ Tạo 2 thư mục con: **data**, **log**
 - ❖ Tạo file ***config.txt***
- ❖ Nội dung file config.txt

```
##store data
```

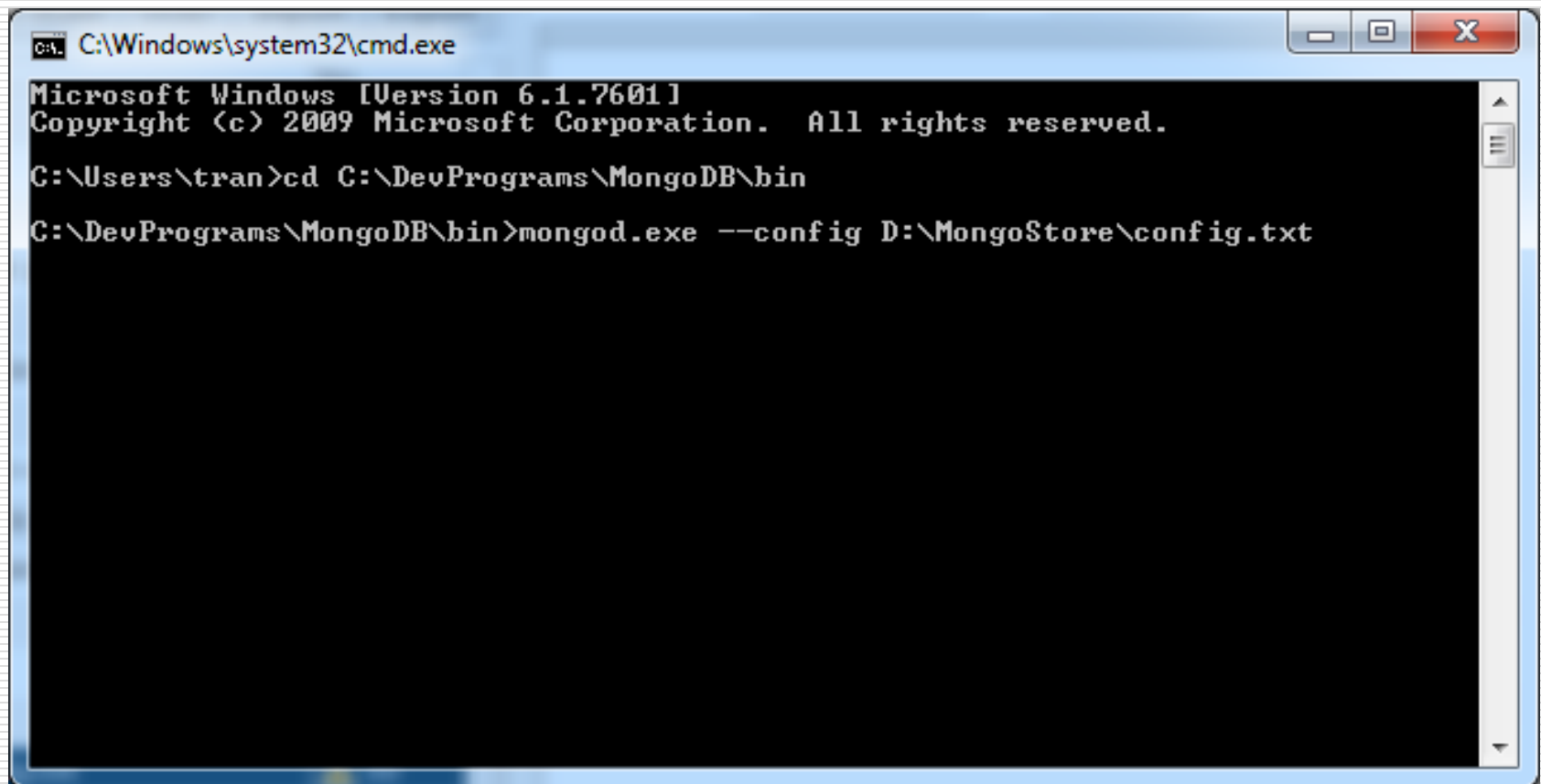
```
dbpath=D:\MongoStore\data
```

```
##all output go here
```

```
logpath=D:\MongoStore\log\mongo.log
```

Chạy MongoDB

- ❖ Chạy cm.exe, CD tới thư mục **bin** của **mongodb**
- ❖ Chạy lệnh: **mongod.exe --config D:\MongoStore\config.txt**

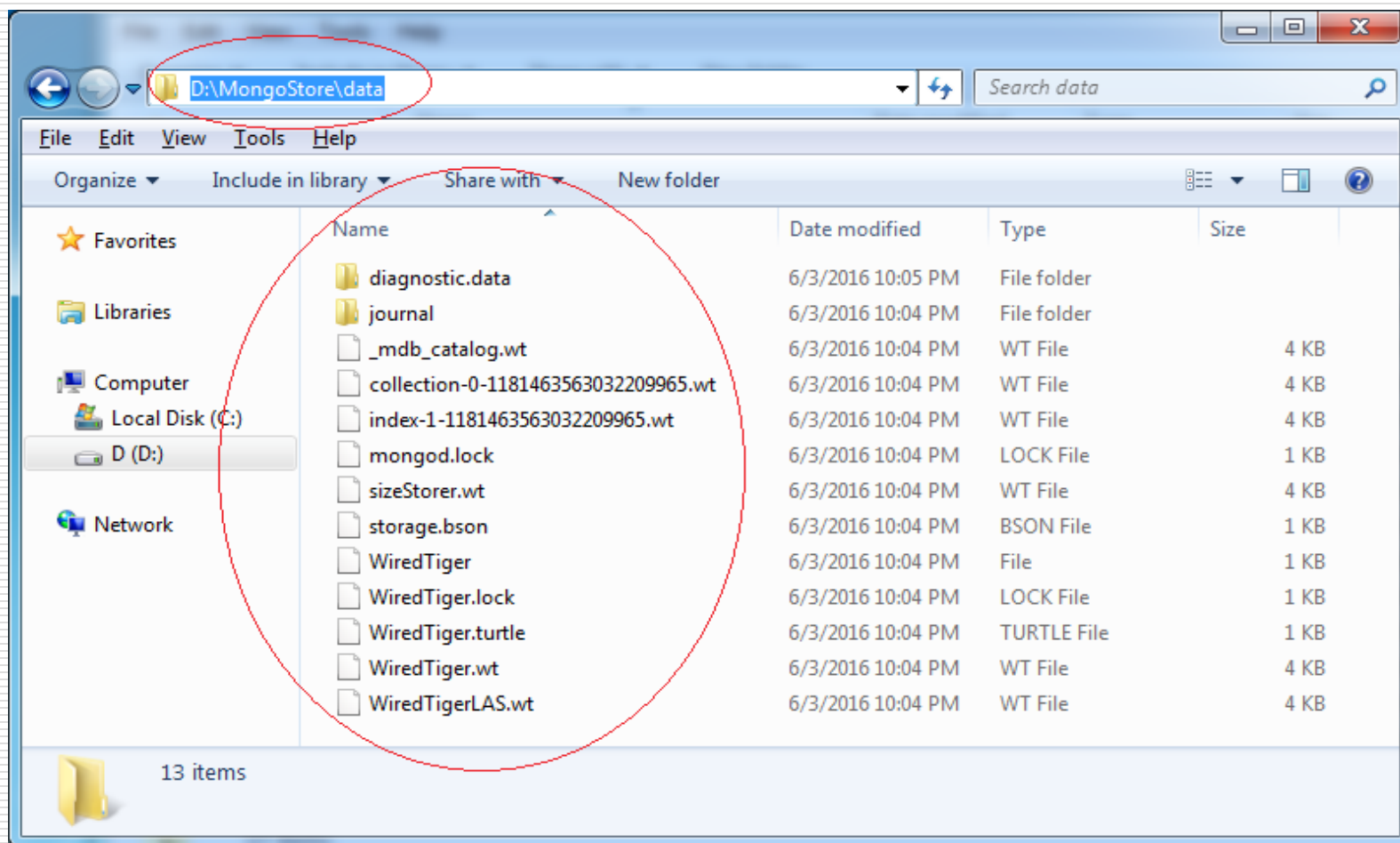


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\tran>cd C:\DevPrograms\MongoDB\bin
C:\DevPrograms\MongoDB\bin>mongod.exe --config D:\MongoStore\config.txt
```

Chạy MongoDB

❖ Kết quả trong thư mục “MongoStore/data”:



Làm việc với MongoDB

❖ Xem trang: <https://vietjack.com>

CSDL Cassandra

- ❖ Giới thiệu về CSDL Cassandra
- ❖ Đặc trưng của CSDL Cassandra

Giới thiệu về CSDL Cassandra

- ❖ Là CSDL NoSQL, được phát triển bởi Facebook vào năm 2007, sau đó, được tặng cho quỹ Apache vào 2/2010
- ❖ Là hệ cơ sở dữ liệu phân tán, dữ liệu được lưu trữ trên nhiều node của nhiều máy khác nhau
- ❖ Ngôn ngữ phát triển Cassandra là Java
- ❖ Được thiết kế có thể chạy trong phần cứng giá rẻ
- ❖ Dữ liệu được lưu vào table, sau đó dùng 1 ngôn ngữ query như SQL để thực hiện thao tác với dữ liệu

Đặc trưng của CSDL Cassandra

- ❖ Tính phân tán và không tập trung (distributed and decentralized): Phân chia dữ liệu thành nhiều phần, đặt trên nhiều node khác nhau
- ❖ Tính mềm dẻo (elastic scalability): Hệ thống có thể dễ dàng mở rộng hoặc giảm bớt số node trong cluster
- ❖ Tính sẵn sàng cao (high availability): Dữ liệu được sao lưu thành nhiều bản và được chia thành nhiều node
- ❖ Tính chấp nhận lỗi (fault tolerance): Do dữ liệu được sao chép thành nhiều bản trên các node của cluster
- ❖ Tính hướng cột (column oriented key-value store)
- ❖ Hiệu năng cao (high performance): Được thiết kế đầy đủ lợi ích cho máy đa luồng/đa lõi, có thể chạy trên hàng chục những máy được đặt trong các trung tâm dữ liệu với quy mô nhất quán và liên tục với hàng trăm terabyte dữ liệu