



## Nội dung chương 1

1. NoSQL là gì
2. Lịch sử phát triển của NoSQL
3. Các đặc trưng của NoSQL
4. So sánh NoSQL và RDBMS
5. Phân loại và sử dụng NoSQL
6. Một số hệ quản trị CSDL NoSQL

# NoSQL



3

Bộ môn HTTT

## 1. NoSQL là gì

- Cơ sở dữ liệu **NoSQL** (tên gốc là "Non SQL" (phi SQL) hoặc "non relational" (phi quan hệ)) cung cấp một cơ chế để lưu trữ và truy xuất dữ liệu được mô hình hóa khác với các quan hệ bảng được sử dụng trong các cơ sở dữ liệu kiểu quan hệ.
- Các hệ thống NoSQL cũng đôi khi được gọi là "**Not only SQL**" (không chỉ là SQL).
- NoSQL ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm.
- Các hệ thống NoSQL không cố gắng bảo đảm tính nguyên tố, nhất quán, tách biệt và bền vững (ACID).

4

Bộ môn HTTT

## 2. Lịch sử phát triển của NoSQL

- Được Carl Strozzi giới thiệu vào năm 1998 để đặt tên cho cơ sở dữ liệu quan hệ mã nguồn mở Strozzi NoSQL (không sử dụng SQL cho việc truy vấn).
- Vào 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. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ database mới: distributed (phân tán) + non-relational (không ràng buộc).

5

Bộ môn HTTT

## Tại sao cần NoSQL?

- Lượng dữ liệu mà các hệ thống cần phải xử lý giờ đây ngày một lớn. Ví dụ như Google, Facebook phải lưu trữ và xử lý một lượng dữ liệu cực lớn mỗi ngày.

Tham khảo:

<https://techjury.net/blog/how-much-data-is-created-every-day/#gref>

6

Bộ môn HTTT

## Database xử lý dữ liệu lớn

- Xuất phát từ yêu cầu thực tế cần những database có khả năng lưu trữ dữ liệu với lượng rất lớn, truy vấn dữ liệu với tốc độ cao.
- Đây là những vấn đề mà các relational database không thể giải quyết được.

7

Bộ môn HTTT

## Dễ dàng thay đổi cấu trúc dữ liệu

<b>id</b>	<b>name</b>	<b>isActive</b>	<b>dob</b>
1	John Smith	true	1977-03-20
2	Sarah Jones	false	1982-10-03
3	Adam Stark	true	2000-07-05

Relation database

Document 1	Document 2	Document 3
<pre>{   "id": "1",   "name": "John Smith",   "isActive": "true",   "dob": "1977-03-20" }</pre>	<pre>{   "id": "2",   "fullName": "Sarah Jones",   "isActive": "false",   "dob": "1982-10-03" }</pre>	<pre>{   "id": "3",   "fullName": {     "first": "Adam",     "last": "Stark"   },   "isActive": "true",   "dob": "2000-07-05" }</pre>

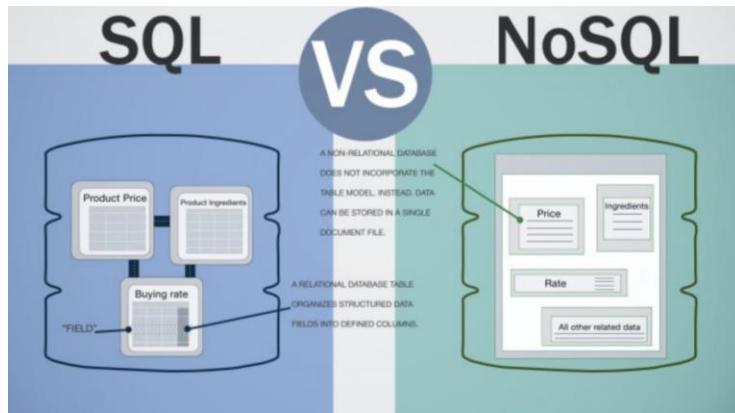
NoSQL database

8

Bộ môn HTTT

### 3. Đặc trưng của NoSQL

- Non-relational: không ràng buộc

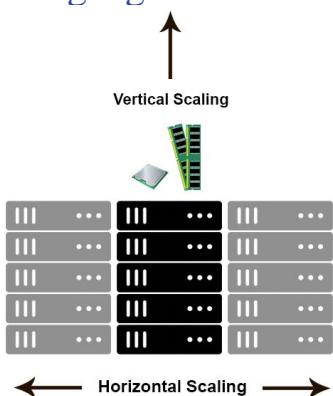


9

Bộ môn HTTT

### Khả năng mở rộng cao

- Vertical scalable: khả năng mở rộng theo chiều dọc
- Horizontal scalable: khả năng mở rộng theo chiều ngang



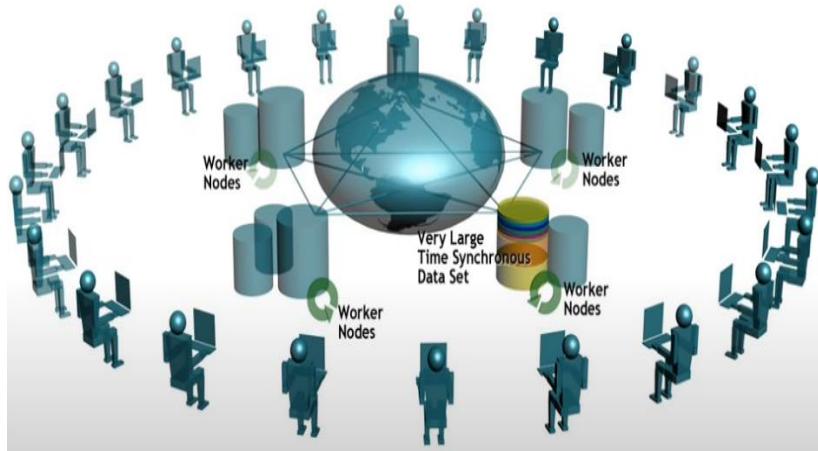
← Horizontal Scaling →

10

Bộ môn HTTT

## Phân tán dữ liệu (Distributed data)

- Distributed Data:

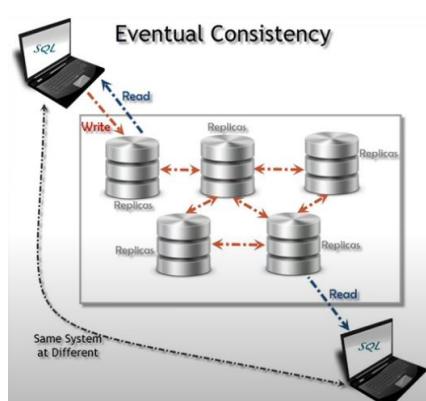


11

Bộ môn HTTT

## Nhất quán cuối (Eventual consistency)

Không cần đồng bộ ngay sau mỗi lần ghi mà chấp nhận những ảnh hưởng theo phương thức lan truyền và sau một khoảng thời gian các 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.

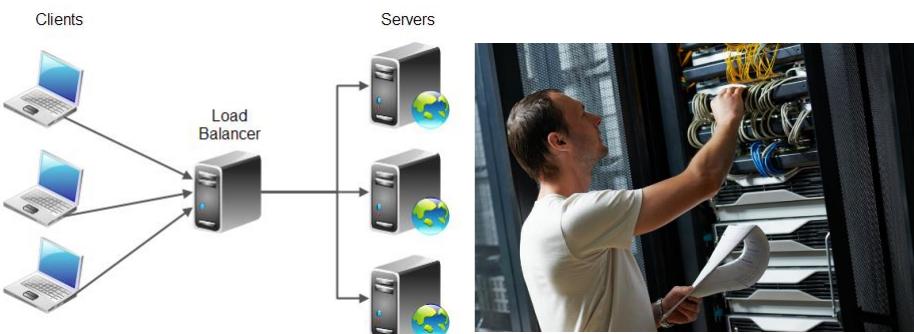


12

Bộ môn HTTT

## Tính sẵn sàng cao (High Availability)

- Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node (commodity machine) nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống.



13

Bộ môn HTTT

## Triển khai linh hoạt (Deployment Flexibility)

- 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 cũng không đòi hỏi cấu hình phần cứng mạnh, đồng nhất.

14

Bộ môn HTTT

## Nhược điểm của NoSQL

- Không có ràng buộc tham chiếu
- Không có ngôn ngữ truy vấn cấu trúc → lập trình nhiều hơn
- Không đảm bảo tính ACID → ít bảo toàn thông tin hơn
- Không dễ dàng tích hợp với các ứng dụng có hỗ trợ SQL khác.

15

Bộ môn HTTT

## Ai sử dụng NoSQL



16

Bộ môn HTTT

## 4. So sánh NoSQL và RDBMS

Đặc điểm	RDBMS	NoSQL
<b>Hiệu suất</b>	Kém hơn NoSQL vì khi truy vấn nó phải tính toán, kiểm tra và xử lý các mối quan hệ giữa các bảng.	Cực tốt Vì nó bỏ qua các ràng buộc dữ liệu
<b>Khả năng mở rộng</b>	Phức tạp hơn nếu đã tồn tại dữ liệu trong database.	Mở rộng dễ dàng
<b>Hiệu suất đọc/ghi</b>	Kém vì phải đảm bảo ràng buộc toàn vẹn	Nhanh hơn vì bỏ qua các ràng buộc giữa các bảng
<b>Thay đổi số node trong hệ thống</b>	Phải shutdown cả hệ thống. Việc thay đổi số node phức tạp.	Không cần phải shutdown cả hệ thống. Việc thay đổi số node đơn giản, không ảnh hưởng đến hệ thống.

17

Bộ môn HTTT

## Dùng NoSQL hay SQL

- Cơ sở dữ liệu SQL sẽ là lý tưởng cho các dự án có yêu cầu đã được xác định sẵn và cần có sự thống nhất chặt chẽ giữa các dữ liệu.
- Cơ sở dữ liệu NoSQL sẽ là lựa chọn tốt cho các yêu cầu chưa được xác định hoàn toàn, dữ liệu không quá liên quan, không cần nhiều sự thống nhất, các dữ liệu chưa được xác định hết và có thể sẽ phát triển ra thêm nhiều hơn, các dự án xem trọng yếu tố tốc độ và tính mở rộng.

18

Bộ môn HTTT

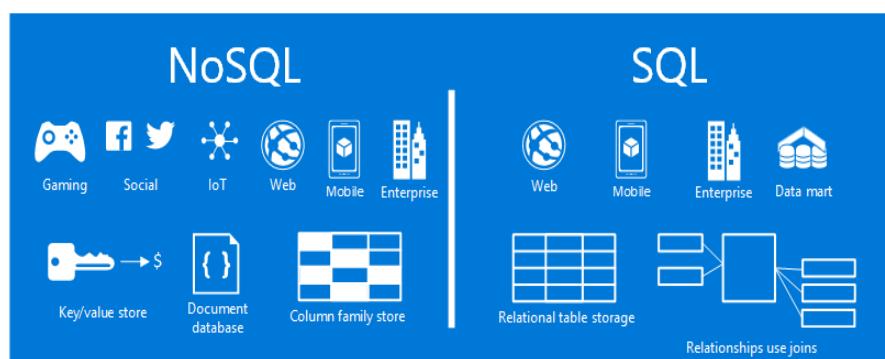
## Dùng NoSQL hay SQL

- NoSQL sẽ phù hợp nhất với các dữ liệu có tổ chức với một yêu cầu mềm lỏng - có thể được thay đổi, mở rộng cho phù hợp mọi hoàn cảnh.
- Các trường hợp hay được sử dụng: Các mạng xã hội, các hệ thống quản lý khách hàng và phân tích trang web.

19

Bộ môn HTTT

## Dùng NoSQL hay SQL



20

Bộ môn HTTT

## 5. Phân loại NoSQL

- **Có thể chia NoSQL thành 4 loại chính sau:**
  - ✓ Key/value store
  - ✓ Column Family store
  - ✓ Graph store
  - ✓ Document store

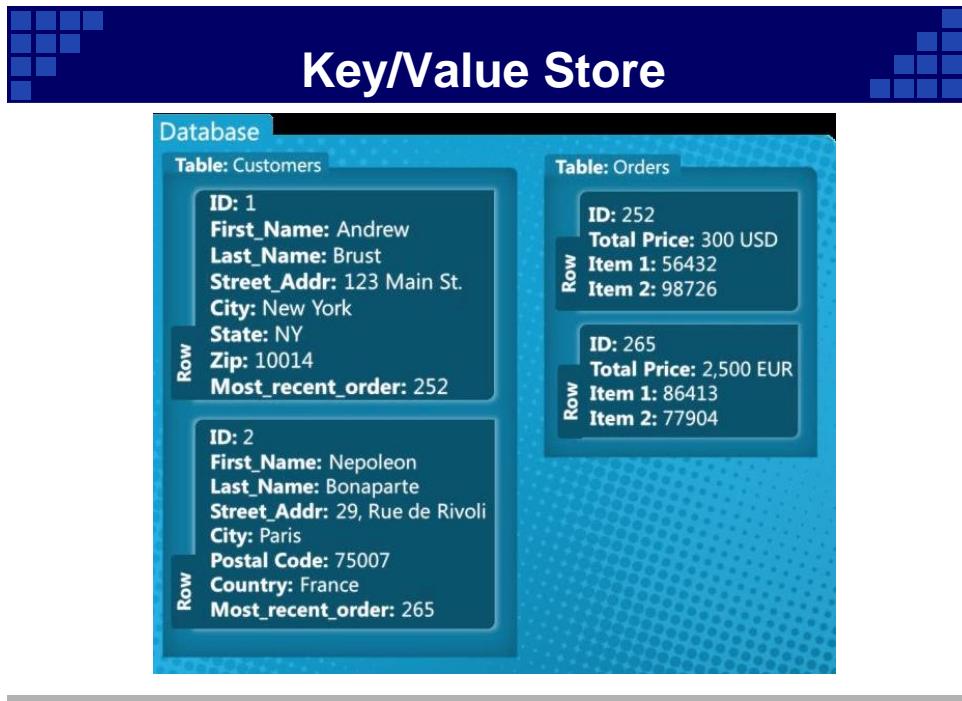
21

Bộ môn HTTT

### Key/Value Store

- **Giới thiệu:** Dữ liệu được lưu trữ trong database dưới dạng key-value. Để truy vấn dữ liệu trong database, ta dựa vào key để lấy ra value. Các database dạng này có tốc độ truy vấn rất nhanh.
- **Database tiêu biểu:** Riak, Redis, MemCache, Project Voldemort, CouchBase, Oracle NoSQL
- **Ứng dụng:** Do tốc độ truy xuất nhanh, key-value database thường được dùng để làm cache cho ứng dụng (Tiêu biểu là Redis và MemCache). Ngoài ra, nó còn được dùng để lưu thông tin trong sessions, profiles/preferences của user...

Bộ môn HTTT



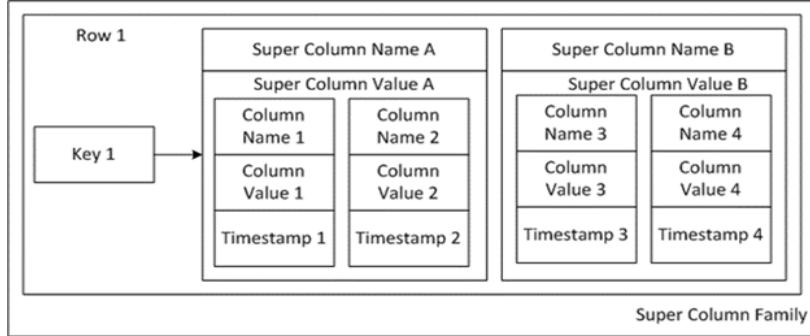
Bộ môn HTT23

## Column Family Store

- Giới thiệu:** Dữ liệu được lưu trong database dưới dạng các cột, thay vì các hàng như SQL. Mỗi hàng sẽ có một key/id riêng. Điểm đặc biệt là các hàng trong một bảng sẽ có số lượng cột khác nhau. Câu lệnh truy vấn của nó khá giống SQL.
- Database tiêu biểu:** Hadoop/Hbase – Apache, BigTable – Google, Cassandra - Facebook, HyperTable – Zvents Inc, Apache HBase
- Ứng dụng:** Column-Family Database được sử dụng khi cần ghi một số lượng lớn dữ liệu, ứng dụng trong big data.

Bộ môn HTT23

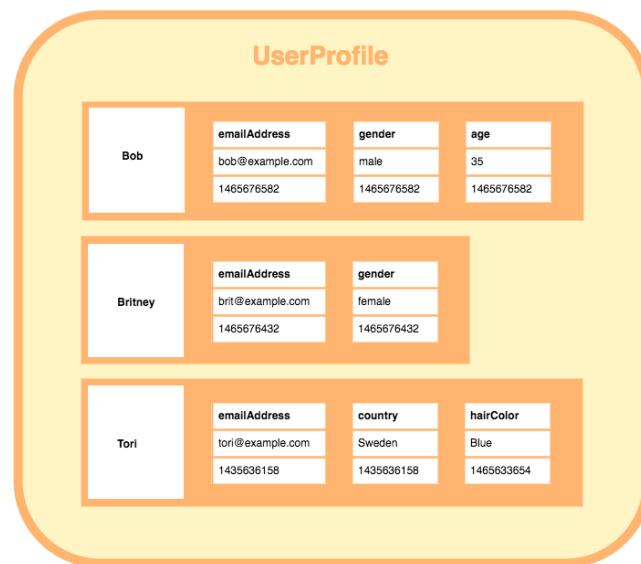
## Column Family Store



25

Bộ môn HTTT

## Column Family Store



26

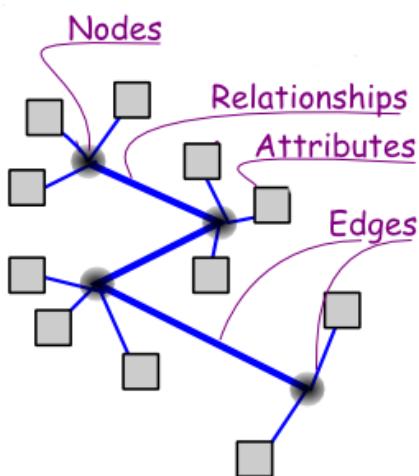
Bộ môn HTTT

## Graph Store

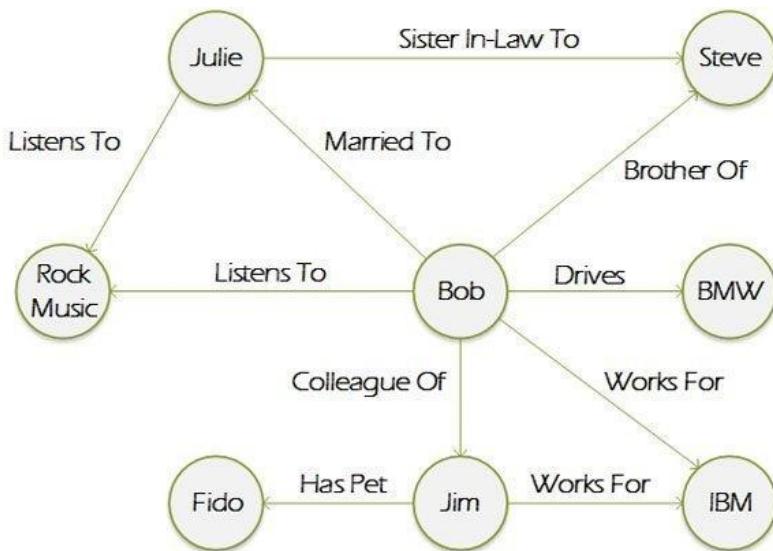
- **Giới thiệu:** Kiểu đồ thị này cho phép lưu trữ các thực thể và quan hệ giữa các thực thể. Các đối tượng này còn được gọi là các nút, trong đó có các thuộc tính. Mỗi nút là một thể hiện của một đối tượng trong ứng dụng. Quan hệ được gọi là các cạnh, có thể có các thuộc tính.
- **Database tiêu biểu:** Neo4j, InfiniteGraph, OrientDB, HYPERGRAPHDB, Sones, AllegroGraph, Core Data, DEX, FlockDB, InfoGrid,..
- **Ứng dụng:** Khi cần truy vấn các mối quan hệ, graph database truy vấn nhanh và dễ hơn nhiều so với database. Nó được dùng trong các hệ thống: mạng nơ ron, chuyển tiền bạc, mạng xã hội (tìm bạn bè), giới thiệu sản phẩm (dựa theo sở thích/lịch sử mua sắm của người dùng)...

Bộ môn HTTT

## Graph Store



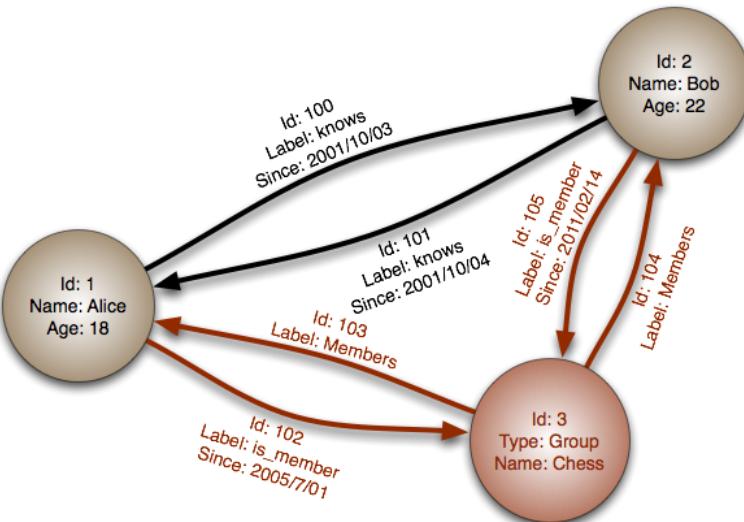
# Graph Store



29

Bộ môn HTTT

# Graph Store



Bộ môn HT30

## Document Store

- **Giới thiệu:** Mỗi object sẽ được lưu trữ trong database dưới dạng một document. Dữ liệu sẽ được lưu trữ dưới dạng BSON/JSON/XML dưới database. Dữ liệu không schema cứng như SQL, do đó ta có thể thêm/sửa field, thay đổi table, ... rất nhanh và đơn giản. Database dạng này có tốc độ truy vấn nhanh, có thể thực hiện các câu truy vấn phức tạp, dễ mở rộng (scalability), mỗi database có một kiểu truy vấn riêng. **Database tiêu biểu:** MongoDB, RavenDB, CouchDB, TerraStone, OrientDB
- **Ứng dụng:** Do nhanh và linh động, document database thường đóng vai trò làm database cho các ứng dụng prototype, big data, e-commerce. Ngoài ra, ta còn dùng nó để lưu log hoặc history.

Bộ môn HTTT

## Document Store

*This sample of JSON side by side with its XML equivalent shows the name-value pairs*

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>
```

32

Bộ môn HTTT

## Document store

33

Name	Producer	Data model	Querying
MongoDB	10gen	object-structured documents stored in collections; each object has a primary key called ObjectId	manipulations with objects in collections (find object or objects via simple selections and logical expressions, delete, update,)
Couchbase	Couchbase	document as a list of named (structured) items (JSON document)	by key and key range, views via Javascript and MapReduce

Bộ môn HTTT



Hết chương 1