

TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN CƠ SỞ DỮ LIỆU NÂNG CAO

ĐỀ TÀI: TÌM HIỂU VỀ MONGODB



Giảng viên hướng dẫn: Ths. Lương Trần Hy Hiến

Sinh viên thực hiện: Nhóm 10

Nguyễn Cảnh Quyết - 43.01.104.146

Trương Chí Lâm - 43.01.104.089

Nguyễn Tấn Tài - 43.01.104.152

Lâm Kiệt Tường - 43.01.104.202

Lê Văn Tiến - 43.01.104.180

Nguyễn Long Hồ - 43.01.104.053

Thành phố Hồ Chí Minh – Năm 2019

❧ Mục lục ❧

<i>I. Tìm hiểu về NoSQL và MongoDB:</i>	2
1. NoSQL :	2
1.1. NoSQL là gì?	2
1.2. NoSQL database là gì?	2
1.3. Tại sao lại cần phải có NoSQL.	3
2. MongoDB :	3
2.1 Ưu điểm MongoDB :	4
2.2 Nhược điểm MongoDB :	5
2.3 Khi nào nên sử dụng MongoDB :	6
2.4 Lịch sử phát triển MongoDB.	6
<i>II. Các thuật ngữ hay sử dụng trong MongoDB:</i>	6
<i>III. Đăng kí để sử dụng MongoDB Cloud:</i>	7
1. Tạo tài khoản MongoDB và Login	7
.....	Error! Bookmark not defined.
2. Tạo Project MongoDB Cloud:	8
.....	14
<i>IV. Sử dụng kết nối và truy vấn dữ liệu MongoDB trên NodeJS :</i>	14
1. Connect tới MongoDB Cloud:	14
.....	15
2. Lệnh dùng trong collection:	15
<i>V. Phần Demo sản phẩm:</i>	18
1. Tạo môi trường làm việc :	18
.....	21
2. Giới thiệu sản phẩm Demo	23
<i>VI. Tổng kết và đánh giá:</i>	25
1. Tổng kết	25
2. Đánh giá	25
TÀI LIỆU KHAM KHẢO.....	26

I. Tìm hiểu về NoSQL và MongoDB:

- Trong những năm gần đây, với sự ra đời và phát triển mạnh mẽ của NoSQL thì MongoDB cũng đang nhận được nhiều sự chú ý trong cộng đồng công nghệ. Điểm mạnh của NoSQL nói chung và MongoDB nói riêng đó là tính linh hoạt trong việc cấu trúc dữ liệu do đó nó giúp đáp ứng tốt với những thay đổi hay việc mở rộng cơ sở dữ liệu. Nếu bạn là người mới nghiên cứu về NoSQL thì đây là một cơ sở dữ liệu đầu tiên nên tiếp cận bởi nó dễ dàng cài đặt và sử dụng để tiếp cận với những ý tưởng lưu trữ mới.

1. NoSQL :

1.1. NoSQL là gì?

- NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL.
- NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.
- NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng,...
- Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc .v.v ...
- NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.
- NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...

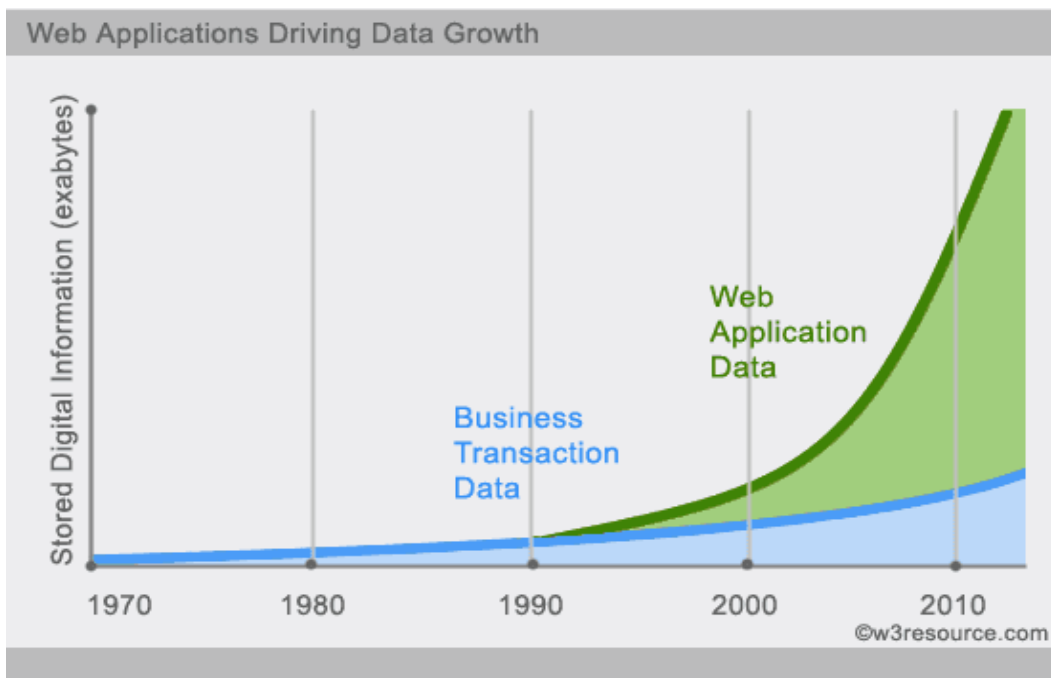
1.2. NoSQL database là gì?

- Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các lightweight open source relational database (cơ sở dữ liệu quan hệ nguồn mở nhỏ) nhưng không sử dụng SQL cho 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 thể hệ database mới: distributed (phân tán) + non-relational (không ràng buộc). Đây là 2 đặc tính quan trọng nhất.

1.3. Tại sao lại cần phải có NoSQL.

- Sở dĩ người ta phát triển NoSQL xuất phát từ yêu cầu cần những database có khả năng lưu trữ dữ liệu với lượng cực lớn, truy vấn dữ liệu với tốc độ cao mà không đòi hỏi quá nhiều về năng lực phần cứng cũng như tài nguyên hệ thống và tăng khả năng chịu lỗi.
- Đây là những vấn đề mà các relational database không thể giải quyết được.
- Lượng dữ liệu mà các hệ thống cần phải xử lý giờ đây ngày 1 lớn. Ví dụ như Google, Facebook phải lưu trữ và xử lý một lượng dữ liệu cực lớn mỗi ngày.



Biểu đồ lưu trữ dữ liệu tăng theo thời gian

2. MongoDB :

- MongoDB (bắt nguồn từ “humongous”) là một hệ cơ sở dữ liệu NoSQL mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể đựng dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data)



Hình 1.2.1: Logo của MongoDB

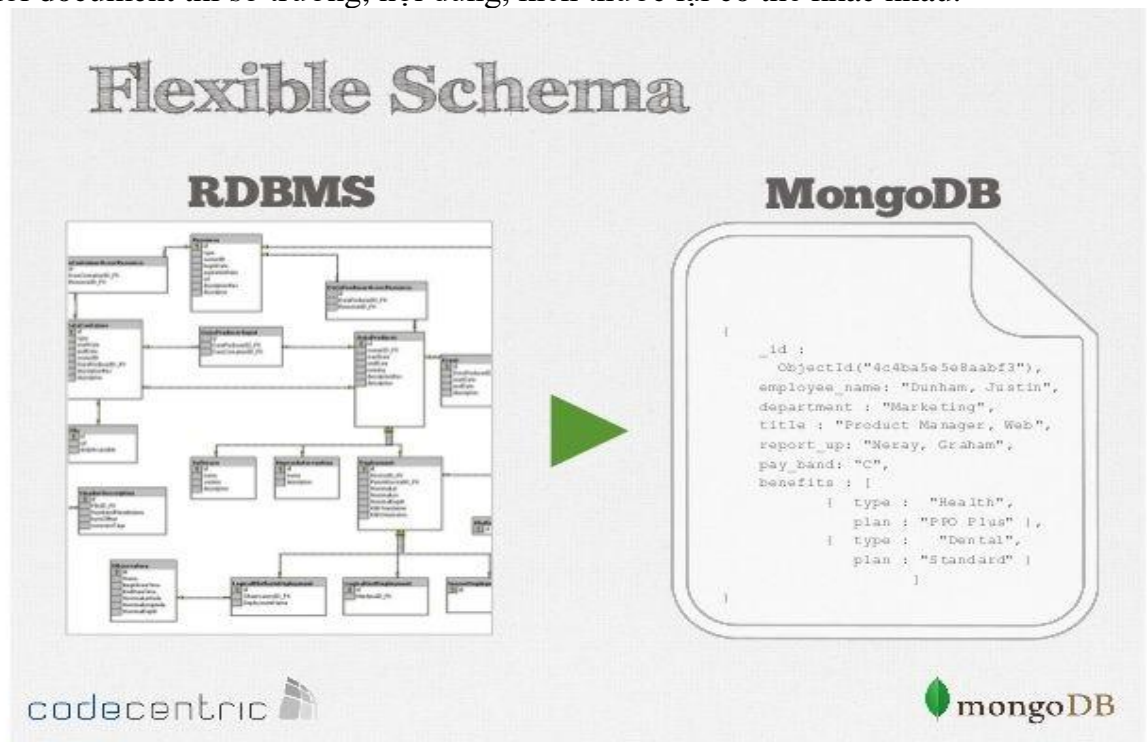
- MongoDB đứng ở tốp đầu trong danh sách các Hệ quản trị cơ sở dữ liệu NoSQL. Thay cho việc lưu trữ dữ liệu vào các bảng có quan hệ với nhau như truyền thống, MongoDB lưu các dữ liệu cấu trúc dưới dạng giống với JSON(JavaScript Object Notation) và gọi tên là BSON

- MongoDB có hai phiên bản Community (miễn phí) và Enterprise (trả phí). Mã nguồn, quản lý phiên bản đặt tại Github [1], phát hành theo giấy phép phần mềm mã nguồn mở GNU Affero General Public License (AGPL) hoặc Apache License tùy theo phiên bản. MongoDB được viết bằng ngôn ngữ lập trình C++ (80,1%), JavaScript (16,5%), Python (2,8%) và các ngôn ngữ lập trình khác (0,6%). MongoDB là hệ quản trị cơ sở dữ liệu chạy trên nhiều hệ điều hành (Linux, Windows, MacOS), trên kiến trúc máy tính 32bit hoặc 64 bit (tuy nhiên không khuyến khích chạy trên máy tính 32bit do kích thước tập tin sẽ bị giới hạn bởi khả năng đánh địa chỉ ô nhớ của hệ điều hành).

2.1 Ưu điểm MongoDB :

- Dữ liệu linh hoạt - MongoDB là document database, dữ liệu lưu dưới dạng JSON, không bị bó buộc về số lượng field, kiểu dữ liệu...
- Ngôn ngữ truy vấn phong phú - MongoDB có sẵn các method để thực hiện create, read, update, delete dữ liệu(CRUD). Không có các fuction joins phức tạp như MySQL.
- Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau như trong Relational Database Management System (RDBMS), khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các bảng liên quan .
- Dữ liệu trong MongoDB được đánh chỉ mục(index) nên truy vấn nhanh gọn.

- Dễ dàng mở rộng hệ thống do có cluster là cụm các node chứa dữ liệu giao tiếp với nhau.
- Hỗ trợ replica set nhằm đảm bảo việc sao lưu và khôi phục dữ liệu.
- Không cần thiết conversion/mapping (chuyển đổi / ánh xạ) các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu.
- Sử dụng bộ nhớ trong để lưu trữ, cho phép truy cập dữ liệu nhanh hơn.
- Sử dụng MongoDB Cloud cho phép nhiều người dùng có thể cùng nhau xây dựng dự án
- Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, để quản lý. Ví dụ như tạo 1 schema tên là Students chẳng hạn thì chỉ có những gì liên quan đến student thì mới được cho vào schema này. Trong khi đó trong MongoDB thì chỉ 1 collection ta có thể chứa nhiều document khác nhau. Với mỗi document thì số trường, nội dung, kích thước lại có thể khác nhau.



Schema của RDBMS và MongoDB

2.2 Nhược điểm MongoDB :

- MongoDB không có các tính chất ràng buộc như trong RDBMS nên dễ bị làm sai dữ liệu
- Không hỗ trợ join giống như RDBMS nên khi viết function join trong code phải tự làm khiến cho tốc độ truy vấn bị giảm.
- Sử dụng nhiều bộ nhớ: do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên sẽ bị dư thừa dữ liệu.

- Bị giới hạn kích thước bản ghi: mỗi document không được có kích thước lớn hơn 16Mb.
- Dùng MongoDB Cloud thì lúc chạy ở local cần có kết nối Internet để truy cập tới DB Cloud

2.3 Khi nào nên sử dụng MongoDB :

- Danh mục sản phẩm thương mại điện tử
- Blog và quản lý các nội dung.
- Hệ thống realtime (thời gian thực) yêu cầu phản hồi nhanh.
- Lưu trữ dữ liệu không gian địa lý.
- Các trang web mạng di động và mạng xã hội.
- Các hệ thống bigdata với yêu cầu truy vấn nhanh.
- Làm search engine

2.4 Lịch sử phát triển MongoDB.

- Dự án MongoDB được bắt đầu triển khai vào tháng 10 năm 2007 bởi công ty phần mềm **10gen** trong khi công ty này đang xây dựng một nền tảng như là dịch vụ (Platform as a Service) giống như Google App Engine. Năm 2009, công ty chuyển sang mô hình phát triển nguồn mở, với công ty cung cấp hỗ trợ thương mại và các dịch vụ khác. Năm 2013, 10gen đổi tên thành MongoDB Inc. Nó được một số tổ chức sử dụng trong thực tế như:

- Caigslis : Công ty làm việc trong lĩnh vực môi giới quảng cáo trên các website khác (giống adMicro của Việt Nam). MongoDB giúp cho công ty này quản lý hàng tỉ các bản ghi quảng cáo thuận tiện và nhanh chóng.
- Foursquare là một mạng xã hội gắn các thông tin địa lý. Công ty này cần lưu trữ dữ liệu của rất rất nhiều vị trí của các địa điểm như quán cafe, nhà hàng, điểm giải trí, lịch sử, ... và ghi lại những nơi mà người sử dụng đã đi qua.
- CERN : Trung tâm nghiên cứu năng lượng nguyên tử của Châu Âu, sử dụng MongoDB để lưu trữ lại các kết quả, dữ liệu thí nghiệm của mình. Đây là một lượng dữ liệu khổng lồ sẽ dùng để sử dụng trong tương lai.

II. Các thuật ngữ hay sử dụng trong MongoDB:

_id: Là trường bắt buộc có trong mỗi document. Trường _id đại diện cho một giá trị duy nhất trong document MongoDB. Trường _id cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một _id đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.

Collection: Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.

Cursor: Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.

Database: Nơi chứa các Collection, giống với cơ sở dữ liệu RDMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một máy chủ MongoDB có thể chứa nhiều Database.

Document: Một bản ghi thuộc một Collection thì được gọi là một Document. Các Document lần lượt bao gồm các trường tên và giá trị.

Field: Là một cặp name – value trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.

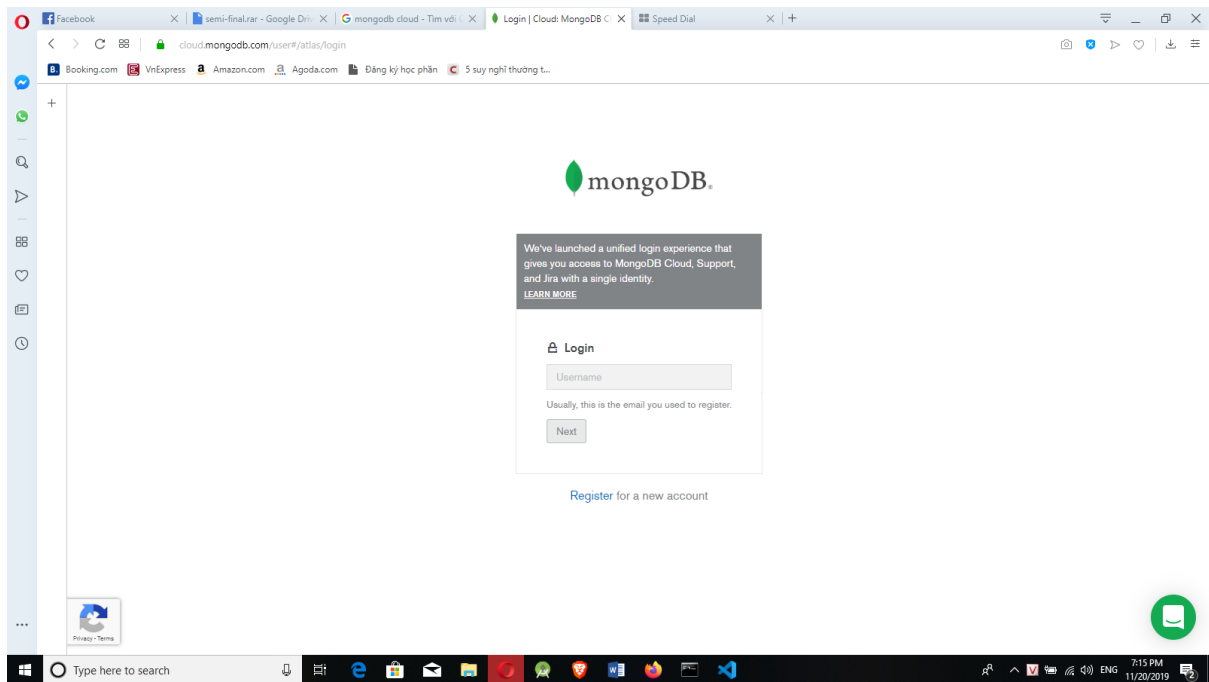
JSON: Viết tắt của JavaScript Object Notation. Con người có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang hỗ trợ rất nhiều ngôn ngữ lập trình.

Index: Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Chỉ số lưu trữ giá trị của một fields cụ thể hoặc thiết lập các fields, sắp xếp theo giá trị của các fields này. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn. Quá trình quét này là không hiệu quả và yêu cầu MongoDB để xử lý một khối lượng lớn dữ liệu.

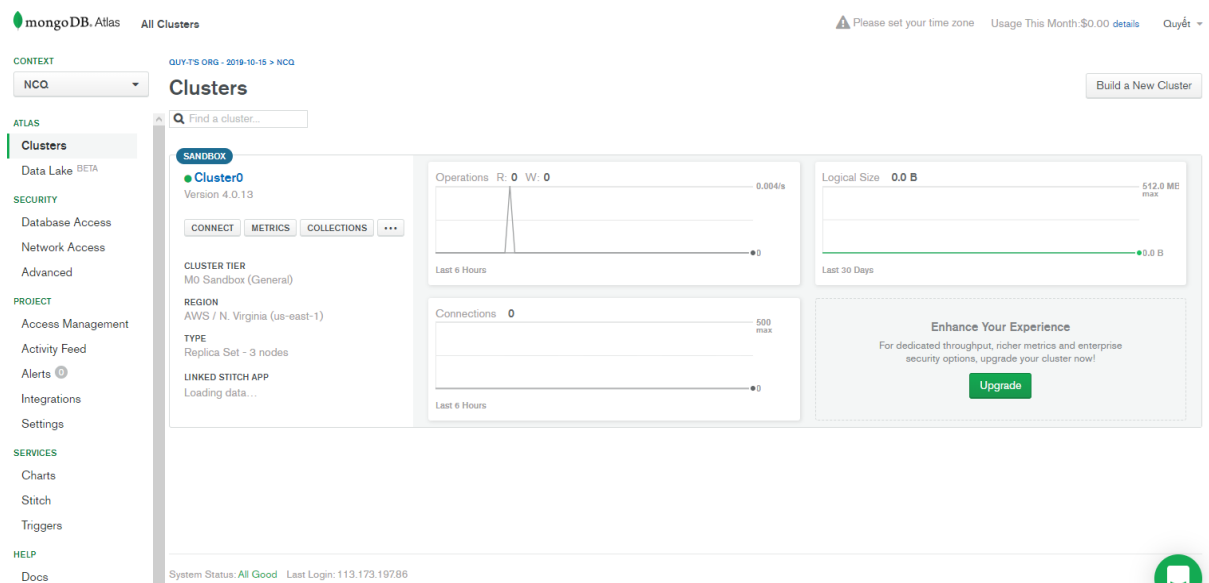
III. Đăng kí để sử dụng MongoDB Cloud:

1. Tạo tài khoản MongoDB và Login

Truy cập : <https://cloud.mongodb.com/user>



Hình 3.1.1 : Login MongoDB Cloud



Hình 3.1.2 : Giao diện MongoDB Cloud

2. Tạo Project MongoDB Cloud:

INTEXT

QUY-TS ORG - 2019-10-15 > PROJECTS

Quy-t's Org - 2...

Create a Project

Organization

Projects

Activity Feed

Access

Alerts 0

Billing

Settings

LP

Docs

Name Your Project Add Members Next

Name Your Project

Project names have to be unique within the organization (and other restrictions).

Project Name

Cancel Next

Hình 3.2.1 : Đặt tên Project

Create a Project

✓ Name Your Project Add Members Go Back Create Project

Add Members and Set Permissions

Invite new or existing users via email address...

Give your members access permissions below.

demon.bmt@gmail.com (you) Project Owner

Cancel Go Back Create Project

Hình 3.2.2 : Thêm thành viên cùng quản lý (Có thể bỏ qua)



Create a cluster

Choose your cloud provider, region, and specs.

[Build a Cluster](#)

Once your cluster is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

Hình 3.2.3 : Thêm Cluster

The screenshot displays the 'Create a cluster' interface with three main options:

- Starter Clusters**
For teams learning MongoDB or developing small applications.
 - ✓ Highly available auto-healing cluster
 - ✓ End-to-end encryption
 - ✓ Role-based access control
 - ✗ No downtime scaling
 - ✗ Network isolation
 - ✗ Realtime performance metrics

Starting at **FREE**

[Create a cluster](#)
- Single-Region Clusters**
For teams building applications that need advanced development and production-ready environments.
 - ✓ Includes all features from Starter Clusters
 - ✓ No downtime scaling
 - ✓ Network isolation
 - ✓ Realtime performance metrics

Starting at **\$0.08/hr***
*estimated cost \$56.94/month

[Create a cluster](#)
- Multi-Region Clusters**
For teams developing world-class applications that require multi-region resiliency or ultra-low latency.
 - ✓ Includes all features from Starter and Single-Region Clusters
 - ✓ Replicate data across multiple regions

Global Clusters ☐

 - ✓ Globally distributed read and write operations
 - ✓ Control data residency at the document level

Starting at **\$0.13/hr***
*estimated cost \$98.55/month

[Create a cluster](#)

Hình 3.2.4 : Chọn gói sử dụng Clusters

Create a Starter Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our [documentation](#).

Cloud Provider & Region

AWS, N. Virginia (us-east-1) ▼



Create a **free tier cluster** by selecting a region with **FREE TIER AVAILABLE** and choosing the **M0** cluster tier below.

★ Recommended region ⓘ

NORTH AMERICA

★ **N. Virginia (us-east-1)** ★
FREE TIER AVAILABLE

★ **Oregon (us-west-2)** ★
FREE TIER AVAILABLE

EUROPE

★ **Ireland (eu-west-1)** ★
FREE TIER AVAILABLE

★ **Frankfurt (eu-central-1)** ★
FREE TIER AVAILABLE

ASIA

★ **Singapore (ap-southeast-1)** ★
FREE TIER AVAILABLE

★ **Mumbai (ap-south-1)** ★
FREE TIER AVAILABLE

AUSTRALIA

★ **Sydney (ap-southeast-2)** ★

FREE

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Back

Create Cluster

Hình 3.2.5 : Lựa chọn server cloud

Clusters

Build a New Cluster

Find a cluster...

SANDBOX

Cluster0
Version 4.0.13

CONNECT METRICS COLLECTIONS ...

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / N. Virginia (us-east-1)

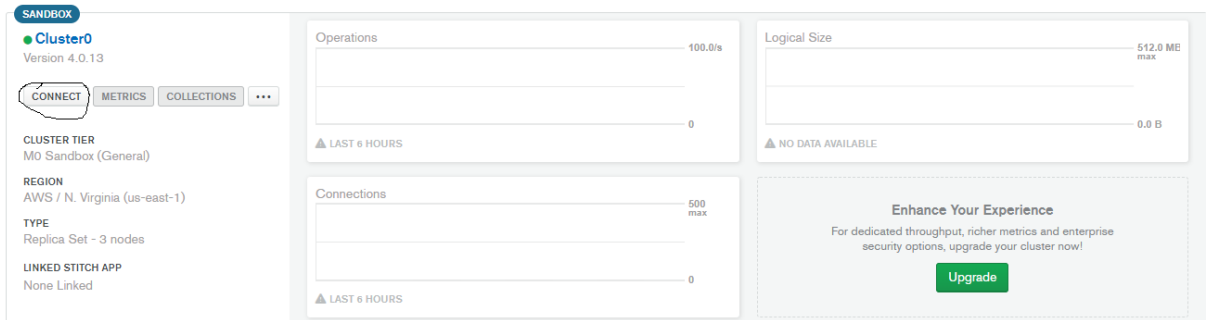
TYPE
Replica Set - 3 nodes

LINKED STITCH APP
None Linked

Your cluster is being created..

New clusters take between 1-3 minutes to provision.

Hình 3.2.6 : Chờ khởi tạo Clusters (1 – 3p)



Hình 3.2.7 : Thực hiện Connect

Connect to Cluster0

Setup connection security > Choose a connection method > Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

You can't connect yet. Set up your firewall access and user security permission below.

1 Whitelist your connection IP address

IP Address	Description (Optional)
42.116.114.80	An optional comment describing this entry

2 Create a MongoDB User

This first user will have [atlasAdmin](#) permissions for this project.
Keep your credentials handy, you'll need them for the next step.

Username <input type="text" value="admin"/>	Password <input type="button" value="Autogenerate Secure Password"/>
<input type="text" value="admin"/>	<input type="text" value="admin"/> <input type="button" value="HIDE"/>

Hình 3.2.8 : Thực hiện Setup Connection Security

Connect to Cluster0


✓ Setup connection security


Choose a connection method


Connect

Choose a connection method [View documentation](#)

See methods to add data and diagnostics in the [Command Line Tools](#) shortcut from within your cluster.

**Connect with the Mongo Shell**
Mongo Shell with TLS/SSL support is required

**Connect Your Application**
Get a connection string and view driver connection examples

**Connect with MongoDB Compass**
Download Compass to explore, visualize, and manipulate your data

Go Back

Close

Hình 3.2.9 : Choose A Connection Method

Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

1 Choose your driver version

DRIVER

Node.js

C

C++

C# / .NET

Java

Node.js

Perl

VERSION

3.0 or later

2

into your application code

Only

Full Driver Example

ssword>@cluster0-v1aja.mongodb.net/test?retryW

Copy

Replace **<password>** with the password for the **admin** user.

When entering your password, make sure that any special characters are **URL encoded**.

Having trouble connecting? [View our troubleshooting documentation](#)

Hình 3.2.10 : Chọn ngôn ngữ thực hiện

Connection String Only

Full Driver Example

mongodb+srv://admin:<password>@cluster0-v1aja.mongodb.net/test?retryW

Copy

Replace **<password>** with the password for the **admin** user.

When entering your password, make sure that any special characters are **URL encoded**.

Hình 3.2.10 : Chuỗi kết nối tới APP

IV. Sử dụng kết nối và truy vấn dữ liệu MongoDB trên NodeJS :

1. Connect tới MongoDB Cloud:

14

```
C:\Users\msi\Desktop\ADB_Final>npm i mongoose
npm WARN adb_final@1.0.0 No description
npm WARN adb_final@1.0.0 No repository field.

+ mongoose@5.7.12
added 19 packages from 13 contributors and audited 152 packages in 3.993s
```

Hình 5.1.1.1 : Thêm Express Mongoose Cho NodeJS

```
final > JS server.js > app
const express = require('express')
const mongoose = require('mongoose')
```

Hình 5.1.1.2 : Thêm Express Mongoose Cho NodeJS

```
mongoose.connect('mongodb+srv://admin:admin@cluster0-xtvbq.mongodb.net/test?retryWrites=true&w=majority', {useNewUrlParser: true})
.then(result => {
  console.log('connect to DB')
})
.catch(error => {
  console.log('***'+error)
})
```

Hình 5.1.2 : Dùng chuỗi kết nối được cung cấp để kết nối tới MongoDB Cloud

2. Lệnh dùng trong collection:

```
const mongoose = require('mongoose')

const authorSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  }
})

module.exports = mongoose.model('Author', authorSchema)
```

Hình 4.2.1 : Cú pháp tạo Schema bằng mongoose

- Tạo Schema hay còn được hiểu là một bảng (My SQL Relational)
- Khai báo các thuộc tính ở trong Schema thông qua type:
 - String
 - Number
 - Date
 - Buffer
 - Boolean

- Mixed
 - ObjectId
 - Array
 - Decimal128
 - Map
- Một số thuộc tính khác:
- Required: kiểu boolean hoặc function, nếu là true sẽ thêm required thuộc tính không được bỏ trống.
 - Default: bất kì kiểu dữ liệu nào, để cài đặt dữ liệu mặc định cho thuộc tính.

```
const Author = require('../models/author')
```

Hình 4.2.2 : Khai báo khi sử dụng model

```
// New Author Route
router.get('/new', (req, res) => {
  | res.render('authors/new', { author: new Author() })
  | })
```

- Router.get : Get đường URL truyền vào / VD :
localhost:3000/author/new
- ⇒ res.render đẩy ra view kèm theo thông tin cần add của author
(res = response) (author chưa có dữ liệu)

```
// Create Author Route
router.post('/', async (req, res) => {
  | const author = new Author({
  |   | name: req.body.name
  |   | })
  | try {
  |   | const newAuthor = await author.save()
  |   | res.redirect(`authors/${newAuthor.id}`)
  | } catch {
  |   | res.render('authors/new', {
  |     | author: author,
  |     | errorMessage: 'Error creating Author'
  |   | })
  | }
  | })
```

Hình 4.2.3 : Cú pháp thêm mới model

- Sau khi Submit tạo một author mới thì dữ liệu từ form được gán vào `name : req.body.name`
- Dùng hàm `save()` của Package Mongooes để lưu đối tượng author mới nhập lên DB

```
router.put('/:id', async (req, res) => {
  let author
  try {
    author = await Author.findById(req.params.id)
    author.name = req.body.name
    await author.save()
    res.redirect(`/authors/${author.id}`)
  } catch {
    if (author == null) {
      res.redirect('/')
    } else {
      res.render('authors/edit', {
        author: author,
        errorMessage: 'Error updating Author'
      })
    }
  }
})
```

Hình 4.2.4 : Cú pháp update model

- Sau khi Submit form Edit sẽ đưa vào URL `/:id`
- Ta sẽ tìm Author thông qua `:id` truyền vào sử dụng `findById()`
- Để lấy được Id trên đường dẫn sử dụng `req.params`. VD: nếu đường dẫn là `/author/:id` -> `req.params.id`

```
router.delete('/:id', async (req, res) => {
  let author
  try {
    author = await Author.findById(req.params.id)
    await author.remove()
    res.redirect('/authors')
  } catch {
    if (author == null) {
      res.redirect('/')
    } else {
      res.redirect(`/authors/${author.id}`)
    }
  }
})
```

Hình 4.2.5 : Cú pháp xóa model

- Giống như update, dữ liệu sẽ xóa theo URL `/:id` truyền vào tìm thông tin author (`findById`) và dùng `remove()` để xóa.

```
router.get('/', async (req, res) => {
  let searchOptions = {}
  if (req.query.name != null && req.query.name !== '') {
    searchOptions.name = new RegExp(req.query.name, 'i')
  }
  try {
    const authors = await Author.find(searchOptions)
    res.render('authors/index', {
      authors: authors,
      searchOptions: req.query
    })
  } catch {
    res.redirect('/')
  }
})
```

Hình 4.2.6 : Cú pháp tìm kiếm

RegExp là một cách để biểu diễn khuôn mẫu của string

```
new RegExp(pattern[, flags])
```

Tham số (flags) i: chỉ ra rằng khuôn mẫu không phân biệt chữ hoa và chữ thường.

Tham số (flags) g: chỉ ra rằng khuôn mẫu áp dụng Global, nghĩa là nó áp dụng cho tất cả thành phần trùng khớp.

V. Phần Demo sản phẩm:

Sản phẩm demo của nhóm sử dụng NoSQL MongoDB Cloud

1. Tạo môi trường làm việc :

https://nodejs.org/en/download/




node
JS

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Downloads

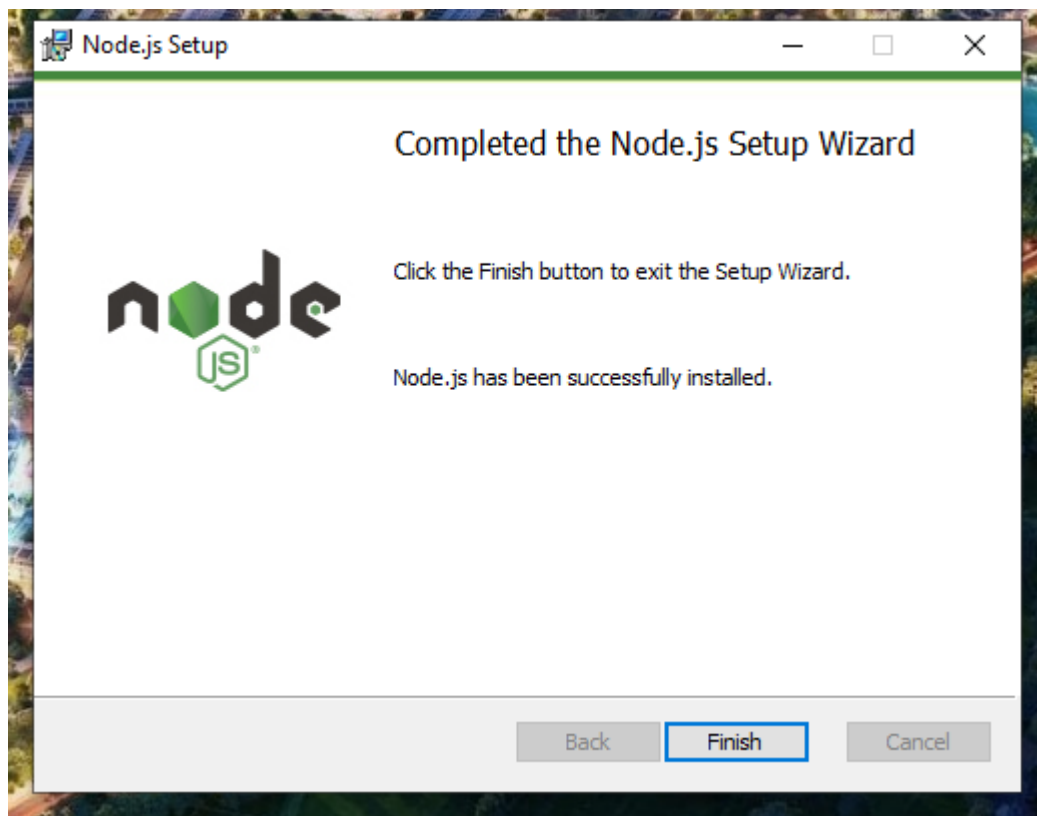
Latest LTS Version: **12.13.1** (includes npm 6.12.1)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer <small>node-v12.13.1-x64.msi</small>	 macOS Installer <small>node-v12.13.1.pkg</small>	 Source Code <small>node-v12.13.1.tar.gz</small>

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.13.1.tar.gz	

Hình 7.1.1 : Download NodeJS



Hình 7.1.2 : Cài đặt vào máy

```
C:\Users\msi\Desktop\ADB_Final>node -v  
v12.13.1
```

Hình 7.1.3 : Kiểm tra version nodeJS : Nếu hiện thì tức là cài thành công

```
C:\Users\msi\Desktop\ADB_Final>npm init
```

Hình 7.1.4.1 : Khởi tạo project

```
C:\Users\msi\Desktop\ADB_Final>npm init  
This utility will walk you through creating a package.json file.  
It only covers the most common items, and tries to guess sensible defaults.  
  
See `npm help json` for definitive documentation on these fields  
and exactly what they do.  
  
Use `npm install <pkg>` afterwards to install a package and  
save it as a dependency in the package.json file.  
  
Press ^C at any time to quit.  
package name: (adb_final) _
```

Hình 7.1.4.2 : Khởi tạo điền các thông tin project

```

save it as a dependency in the package.json file.

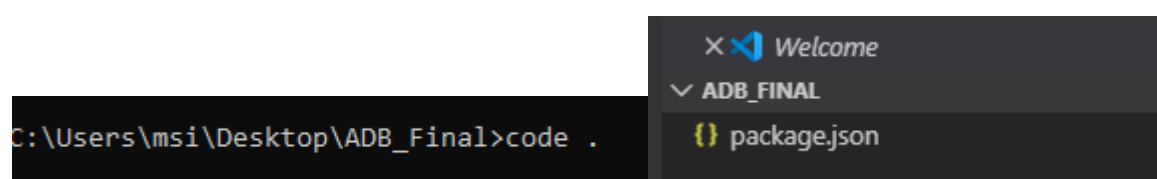
Press ^C at any time to quit.
package name: (adb_final)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\msi\Desktop\ADB_Final\package.json:

{
  "name": "adb_final",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes
C:\Users\msi\Desktop\ADB_Final>

```

Hình 7.1.4.3 : Xác nhận khởi tạo project



Hình 7.1.5 : Project JS được khởi tạo

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

npm i express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN adb_final@1.0.0 No description
npm WARN adb_final@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 126 packages in 6.842s
found 0 vulnerabilities
```

Hình 7.1.6.1 : Import các Package cần thiết

```
C:\Users\msi\Desktop\ADB_Final>npm i mongoose
npm WARN adb_final@1.0.0 No description
npm WARN adb_final@1.0.0 No repository field.

+ mongoose@5.7.12
added 19 packages from 13 contributors and audited 152 packages in 3.993s
```

Hình 7.1.6.2 : Import các Package cần thiết

```
const express = require('express')
const app = express()
app.listen(3000, () => {
  console.log('Listening to port 3000')
})
```

Hình 7.1.7 : Khai báo và bắt đầu code

```
const authorSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  }
})

module.exports = mongoose.model('Author', authorSchema)
```

Hình 7.1.8 : Khởi tạo Schema (Model)

```

router.get('/', async (req, res) => {
  let searchOptions = {}
  if (req.query.name != null && req.query.name !== '') {
    searchOptions.name = new RegExp(req.query.name, 'i')
  }
  try {
    const authors = await Author.find(searchOptions)
    res.render('authors/index', {
      authors: authors,
      searchOptions: req.query
    })
  } catch {
    res.redirect('/')
  }
})

```

Hình 7.1.9 : Chia nhánh xử lý

2. Giới thiệu sản phẩm Demo



Hình 6.2.1 : Trang chủ DEMO

MY LIBRARY

AUTHORS
ADD AUTHOR
BOOKS
ADD BOOK

SEARCH BOOK

Title

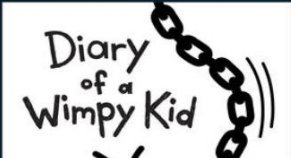
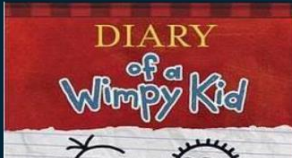
Published After

mm/dd/yyyy

Published Before

mm/dd/yyyy

Search

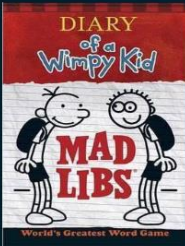



Hình 6.2.2 : Trang tìm kiếm sách

MY LIBRARY

AUTHORS
ADD AUTHOR
BOOKS
ADD BOOK

DIARY OF A WIMPY KID MAD LIBS



AUTHOR: Mad Libs

PUBLISH DATE: Tue Sep 08 2015

PAGE COUNT: 48

DESCRIPTION: Fans of the Diary of a Wimpy Kid book series will go crazy for our brand new Mad Libs, featuring 21 fill-in-the-blank stories starring Greg, Rowley, Manny, and the rest of the gang!

Edit

Delete

View Author

Hình 6.2.3 : Trang xem thông tin chi tiết sách

24

The image shows a web form titled "NEW BOOK" on a dark blue background. The form contains several input fields and buttons. At the top, the title "NEW BOOK" is in white capital letters. Below it, there are two rows of fields. The first row has "Title" and "Author". The "Author" field has a dropdown menu with "Jeff Kinney" selected. The second row has "Publish Date" and "Page Count". The "Publish Date" field has a placeholder "mm/dd/yyyy". Below these, there are two more fields: "Cover" and "Description". The "Cover" field has a placeholder "Drag & Drop your files or [Browse](#)". The "Description" field is a large text area. At the bottom right, there are two buttons: "Cancel" (orange) and "Create" (blue).

Hình 6.2.4 : Trang thêm sách mới

VI. Tổng kết và đánh giá:

1. Tổng kết

- MongoDB không sử dụng mô hình quan hệ ràng buộc(non-relational)
- Mô hình Lưu trữ dạng {"key : value"}
- MongoDB là 1 CSDL phân tán lưu trữ dữ liệu trên nhiều server khác nhau, đa dạng hóa dữ liệu, là dạng dữ liệu lớn(BigData).
- Độ chịu tải cao ,chịu lỗi cao
- Có khả năng nâng cấp theo chiều ngang, dễ dàng bảo trì, thay đổi,sửa chữa , phát triển

2. Đánh giá

- Chính vì những đặc điểm đó ta nên dùng nó cho các hệ thống, ứng dụng đòi hỏi lượng người dùng và dữ liệu lưu trữ đa dạng, lớn đến cực lớn.Ví dụ như mạng xã hội(facebook, Instagram, twitter...), các trang lưu trữ dữ liệu lớn như google, github,

TÀI LIỆU KHAM KHẢO

- [1]. **Ebook NoSQL** - Nhữ Đình Thuận
- [2]. **Kristina Chodorow - Micheal Dirolf, MongoDB: The Definitive Guide**, O'reilly, 2010
- [3]. Eeclo Plugge, Peter Membrey and Tim Hawkins, **The Definitive Guide to MongoDB the NoSQL Database for Cloud and Desktop Computing.**
- [4]. **NoSQL resources** - <http://nosql-database.org>
- [5]. **MongoDB wikipedia** - <https://en.wikipedia.org/wiki/MongoDB>
- [6]. <https://www.mongodb.com>
- [7]. <https://cloud.mongodb.com>
- [8]. <http://kienthucweb.net/gioi-thieu-ve-co-so-du-lieu-mongodb.html>
- [9]. <https://vietjack.com/mongodb/>
- [10]. <https://freetuts.net/tong-quan-ve-mongodb-203.html>