BỘ CÔNG THƯƠNG TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM KHOA CÔNG NGHỆ THÔNG TIN

\$0880A



ĐỒ ÁN TỐT NGHIỆP

ĐÈ TÀI: TÌM HIỂU MONGODB

Giảng viên hướng dẫn : Nguyễn Thị Thanh Thủy

Sinh viên thực hiện :

2001110117 – TRẦN TRUNG HIẾU

2001110099 - TRƯƠNG QUỐC KΫ́

TP. HÔ CHÍ MINH – 2015

LỜI CẨM ƠN

LỜI MỞ ĐẦU

DANH MỤC VIẾT TẮT

DANH MỤC HÌNH ẢNH

DANH MỤC BẨNG

MỤC LỤC

LỜI CÁM ƠN		1
LỜI MỞ ĐẦU		2
DANH MỤC VIẾT	TẮT	3
DANH MỤC HÌNI	H ÅNH	4
DANH MỤC BẢN	G	5
MŲC LŲC		6
CHƯƠNG 1. TỔN	IG QUAN	1
1.1. Đặt vấ	n đề	1
1.2. Muc ti	êu đề tài	1
1.3. Nội du	ıng cσ bản	1
1.4. Phân c	ông công việc	2
1.5. Chức r	năng phần mềm quản lý công việc:	3
CHƯƠNG 2. TÌM	HIỂU CƠ SỞ DỮ LIỆU NOSQL	3
2.1. Khái n	iệm NoSQL	3
2.2. Ưu như	ược điểm của CSDL NoSql	4
2.3. So sán	h NoSql với CSDL quan hệ	5
2.4. Phân le	oại NoSQL	7
CHƯƠNG 3. HỆ (QUẢN TRỊ CƠ SỞ DỮ LIỆU MONGODB	10
1. Cơ sở dữ	liệu MongoDB	10
1.1. Giới	i thiệu MongoDB	10
1.2. Cài	đặt MongoDB	11
1.2.1.	Giới thiệu về MongoDB trên Windows	11
1.2.2.	Một số chú ý khi cài đặt MongoDB trên Windows	
1.2.3.	Hướng dẫn tải về Windows	12
1.2.4.	Hướng dẫn cài đặt trên Windows	13
1.2.5.	Cấu hình MongoDB trên Windows	17
1.2.6.	Bắt đầu làm việc với MongoDB	18
1.2.7.	Công cụ RoboMongo làm việc với MongoDB	19
1.3. Mor	ngoDB CRUD	19
1.3.1. G	iới thiệu MongoDB CRUD	19

ĐỂ TÀI: TÌM HIỂU MONGODB

1.3.2.	Các khái niệm trong MongoDB CRUD	20
1.3.3.	Hướng dẫn MongoDB CRUD	21
1.4. I	Data Model	26
1.4.1.	Giới thiệu Data Model	26
1.4.2.	Các khái niệm cơ bản về Data Model	26
1.4.3.	. Ví dụ và Patterns về Data Model	29
1.4.4.	Tài liệu tham khảo về Data Model Error! Bookmar	k not defined.
1.5. A	Adminitration	29
1.6.	Security	29
1.7.	Aggregation	29
1.8. I	Indexes	29
1.9. I	Replication	29
1.10.	Sharding	30
2. Một s	số đặc trung của Mongodb	30
3. Sử dụ	μng cơ bản một số lệnh trong MongoDB	31
4. Một s	số lệnh truy vấn tương đương giữa SQL và MongoDB	31
CHƯƠNG 4. X	XÂY DỰNG ÚNG DỤNG VỚI MONGODB	1
4.1. Gid	ới thiệu ứng dụng	1
4.2. Thi	iết kế và tạo CSDL trên Mongodb	1
4.3. Xâ	y dựng ứng dụng kết nối Mongodb	1
CHƯƠNG 5. I	KẾT LUẬN	4
TÀI LIỀU THA	NM KHẢO	5

D

Н

CHƯƠNG 1. TỔNG QUAN

1.1. Đặt vấn đề

Cùng với sự phát triển của xã hội hiện nay, đặt biệt là sự bùng nổ về công nghệ thông tin thì vấn đề dữ liệu cần một được đặc biệt chú ý. Từ khi sự xuất hiện của Web 2.0 thì vấn đề xử lý dữ liệu đặc biệt được quan tâm đến, sự ra đời của Web 2.0 đánh dấu một sự phát triển rất lơn mạnh về mặt dữ liệu và các thông tin từ nhưng dữ liệu đó, bởi vì Web 2.0 là tự do về dữ liệu. Nó cho phép các API, RSS, ... có thể truy xuất dữ liệu từ nhiều nguồn khác nhau mà không cần trực tiếp vào trang web đó. Bên cạnh đó sự ra bùng nổ của Web 2.0 và sự phát triển của Internet và số lượng người dùng tăng lên rất lớn, người dùng có thể thỏa mái tạo nội dung trên Web, làm cho dữ liệu trở nên khổng lồ vượt quá giới hạn xử lý của các hệ quản trị cơ sở dữ liệu quan hệ truyền thống.

Để đáp ứng nhu cầu lưu trữ, xữ lý trên dữ liệu không lồ, cần một cơ sở dữ liệu có thể lưu trữ dữ liệu với số lượng lớn và có thể truy xuất một cách nhanh chóng và hiệu quả.

1.2. Muc tiêu đề tài

Trong phạm vi đề tài chúng em tìm hiểu cơ bản khái niệm cơ sở dữ liệu NoSQL là gì, biết được ưu nhược điểm của CSDL NoSql, phân biệt được các loại CSDL NoSQL và so sánh sự giống và khác nhau giữa CSDL NoSql với CSDL quan hệ. Tìm hiểu hệ quản trị CSDL Mongodb, biết cách cài đặt hệ và sử dụng Mongodb để lưu trữ dữ liệu (Tạo database, thêm/ xóa/ sữa và truy vấn dữ liệu).

Biết cách xây dựng ứng dụng sử dụng Mongodb để làm CSDL cho ứng dụng.

1.3. Nội dung cơ bản

- ♣ Phần 1: Tổng quan
 - ✓ Tìm hiểu tổng quan về đề tài
 - ✓ Mục tiêu đề tài
 - ✓ Lên nội dung cơ bản
 - ✓ Phân công công việc
 - ✓ Viết báo cáo
- 4 Phần 2: NoSql
 - ✓ NoSql là gì
 - ✓ Uu nhược điểm của NoSql
 - ✓ So sánh NoSql và CSDL quan hệ
 - ✓ Phân loại NoSql
 - ✓ Viết báo cáo
- ♣ Phần 3: Tìm hiểu Mongodb

- ✓ Cài đặt Mongodb
- ✓ Các lệnh tạo database
- ✓ Các lệnh thêm, xóa, sửa CSDL
- ✓ Truy vấn dữ liệu
- ✓ Khái niệm NoSQL
- ✓ Bảng, cột
- ✓ Ưu điểm
- 4 Phần 3: Kết nối CSDL Mongodb sử dụng C#
 - ✓ Thêm, xóa, sửa CSDL.
 - ✓ Truy vấn dữ liệu
 - ✓ Cài đặt IIS
 - ✓ Tao thư mục để IIS và Webservice trỏ vào.
 - ✓ Code chức năng
 - ✓ Thiết kế layout bằng photoshop
 - ✓ Chuẩn bị resource cho app
- 1.4. Phân công công việc
 - **♣** Thành viên nhóm:
 - ✓ Trần Trung Hiếu
 - ✓ Trương Quốc Kỷ

Thời gian	Nội dung chính	Thời gian	Chi tiết	Người làm
	Tổng quan		Tìm hiểu tổng quan về đề tài	
			Mục tiêu đề tài	
			Lên nội dung cơ bản	
			Phân công công việc	
			Viết báo cáo	
7	NoSql		NoSql là gì	
			Ưu nhược điểm của NoSql	
			So sánh NoSql và CSDL quan	
			hệ	
			Phân loại NoSql	
			Viết báo cáo	
	Mongodb		Tìm hiểu Mongodb	
			Cài đặt Mongodb	
			Các lệnh tạo database	
			Các lệnh thêm, xóa, sửa CSDL	
			Truy vấn dữ liệu	
5	Tính năng của		Khái niệm NoSQL	
Mongodb			Bảng, cột	

		Ưu điểm
5	Ứng dụng C#	Kết nối CSDL Mongodb sử
		dụng C#
		Thêm, xóa, sửa CSDL
		Truy vấn dữ liệu
5	Webservice	Cài đặt IIS
		Tạo thư mục để IIS và
		Webservice trỏ vào.
		Code chức năng
5	Ứng dụng	Thiết kế layout bằng
	Android	photoshop
		Chuẩn bị resource cho app

1.5. Chức năng phần mềm quản lý công việc:

- Người dùng nhập công việc hàng ngày, có đồ ưu tiên của mỗi công việc.
- O Nhắc nhỏ công việc sắp tới trong ngày mai.
- o Mời user khác vào công việc của mình.
- O Cho phép add user vào nhóm.
- O Chat trực tuyến để phục vụ công việc họp nhóm khẩn cấp online
- Người quản lý có thể xem tiến độ công việc và xem thống kê những người chưa hoàn thành, làm chậm để bổ sung người thích hợp vào vị trí đó.
- Đơn giản hóa việc quản lý (người tạo ra công việc chỉ cần quản lý người quản lý và cộng sự (có thể không có)).
- Cho phép thay đổi thời gian làm việc, kéo dài ra hoặc rút ngắn lại.
- Đánh giá được năng lực của những người làm công việc thông qua mức độ công việc hoàn thành như thế nào.
- Tự động cập nhật mức độ công việc gốc khi có bất kỳ người nào làm công việc này sửa mức độ công việc mà họ đang làm.
- Cho phép thêm ghi chú hoặc gợi ý cho những người làm công việc được phân công.

CHƯƠNG 2. TÌM HIỀU CƠ SỞ DỮ LIỆU NOSQL

2.1. Khái niệm NoSQL

NoSQL là Non-Relational: tạm dịch là không có quan hệ. NoSQL là cơ sở dữ liệu không quan hệ, ràng buộc giữa các Collection (hay còn gọi là bảng trong cơ sở dữ liệu bình thường). Tức là giữa các Collection (Bảng) sẽ không có khóa chính, khóa ngoại như trong cơ sở dữ liệu bình thường. Việc này sẽ giảm bớt rối

khi cơ sở dữ liệu chúng ta có nhiều bảng. NoSQL đặc biệt nhấn mạnh đến mô hình lưu trữ cặp giá trị - khóa và hệ thống lưu trữ phân tán.

2.2. Ưu nhược điểm của CSDL NoSql

- a. Ưu điểm của NoSQL:
 - Là mã nguồn mở: không phải mất chi phí và có xu hướng tin cậy, an ninh và nhanh hơn để triển khai so với các hệ quản trị cơ sở dữ liệu độc quyền.
 - ii. Linh hoạt trong việc mở rộng và phát triển: Về phía công ty quản lý thì giúp dễ dàng mở rộng máy chủ khi dữ liệu càng ngày càng lớn hoặc lượng truy cập, tải dữ liệu quá lớn, thay vì thuê một máy chủ lớn hơn để thế máy chủ trước thì công ty chỉ cần thuê thêm một máy chủ khác. Về phía người lập trình thì giúp dễ dàng thêm Collection (Bảng) hoặc cặp dữ liệu (Cột). Dễ dàng trong việc thống kê, truy vấn nhanh.
 - iii. Áp dụng được công nghệ điện toán đám mây: dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ như là Amazon EC2. Giống như tất cả công nghệ đám mây, EC2 dựa vào ảo hóa. Liên kết yếu của ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các các kết nối mạnh. NoSQL lưu trữ dữ liệu thường được mở rộng phạm vi theo chiều ngang tận dụng được sự cung cấp mềm dẻo của đám mây. Giúp mở rộng dữ liệu dễ dàng hơn.
 - iv. Được các hãng lớn sử dụng như Amazon, BBC, Facebook và Google. Làm nền tảng, cơ sở cho các công ty lớn áp dụng cũng như có sự tin cậy cao về hệ quản trị NoSQL.

b. Nhược điểm của NoSQL:

- i. Hỗ trợ không đồng đều cho các doanh nghiệp. Các doanh nghiệp vừa được sự hỗ trợ tốt nhất từ nhà cung cấp RMBMS (vì được phát triển trước NoSQL một thời gian khá dài) còn các doanh nghiệp nhỏ thì thường sử dụng các mã nguồn mở thì không được sự hỗ trợ tốt nhất.
- ii. Chưa được sử dụng rộng rãi vì NoSQL vẫn chưa nhận được sự tin cậy với nhiều doanh nghiệp, một phần dữ liệu đã được xây dựng từ lâu, nên việc chuyển đổi cũng là vấn đề khó với nhiều doanh nghiệp và chưa được hỗ trợ tốt về chức năng cũng như sự ổn định như RMBMS.
- iii. Còn mới lạ với một số lập trình viên. Chưa được sử dụng để đào tạo rộng rãi. Chưa có tool hỗ trợ giao diện tương tác cũng như các phương thức tốt nhất. Dẫn đến hạn chế về tri thức nghiệp vụ.

- iv. Chia sẻ dữ liệu chưa theo một tiêu chuẩn chung. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng API riêng của mình. Sự thiếu hụt các tiêu chuẩn có nghĩa là nó không có khả năng để chuyển một cách đơn giản từ một nhà cung cấp này sang một nhà cung cấp khác.
- 2.3. So sánh NoSql với CSDL quan hệ

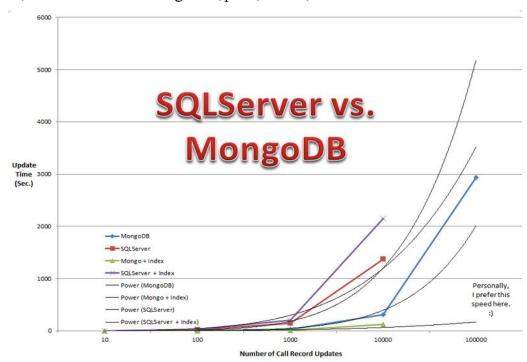
Tiêu chí đánh giá	RDBMs	NoSql
Đánh chỉ mục lượng	Có nhiều điểm yếu kém.	Thực hiện khá tốt.
lớn dữ liệu, phân trang		
hoặc phân phối luồng		
dữ liệu media	m 11	W WI C I
Mô hình dữ liệu	Table	Key/Value, Column-
		based – Tabular,
		Document-based, Graph-
Vân sân ab ân sứm s sho	Vân alim alim arma mán	based data-stores.
Yêu câu phân cứng cho	Yêu câu phân cứng máy chủ cao.	Không nhất thiết phần
dữ liệu lớn	chu cao.	cứng phải cao, có thể mở rộng máy chủ ở nhiều
		nơi với tài nguyên phần
		cứng thấp.
Cách lấy dữ liệu từ	Thông qua chuỗi kết nối	Thông qua API
client.	(tên máy chủ, tên cơ sở	(Application
	dữ liệu, tên tài khoản và	Programming Interface).
	mật khẩu).	11081
Hiệu xuất	Kém hơn	Cực tốt
	SQL	Bỏ qua SQL
	Relational giữa các bảng	Bỏ qua các ràng buộc dữ
		liệu
Khả năng mở rộng	Hạn chế về lượng	Hỗ trợ lượng lớn cho các
		node
Phù hợp với công nghệ	Không	Có.
điện toán đám mây	TT	
Hiệu suất đọc ghi	Kém do thiết kế để đảm	Tốt với mô hình xử lý lô
	bảo sự vào/ra liên tục	và những tối ưu về đọc-
T1 #å: - á 1 4	của dữ liệu	ghi dữ liệu.
Thay đôi số node trong	Phải shutdown cả hệ	Không cần phải
hệ thống	thống. Việc thay đổi số	shutdown cả hệ thống.
	node phức tạp.	Việc thay đổi số node
		đơn giản, không ảnh hưởng đến hệ thống.
Hỗ trợ lập trình hướng	Kém	Tốt
đối tượng	IXCIII	101
doi tuọng		

Tính tiện dụng về mặt	Tốt	Còn chưa hoàn thiện
hỗ trợ		

- a. Các RDBMs hiện tại đã bộc lộ những yếu kém như việc đánh chỉ mục một lượng lớn dữ liệu, phân trang, hoặc phân phối luồng dữ liệu media (phim, ảnh, nhạc ...). Cơ sở dữ liệu quan hệ được thiết kế cho những mô hình dữ liệu nhỏ thường xuyên đọc viết trong khi các Social Network Services lại có một lượng dữ liệu cực lớn và cập nhật liên tục do số lượng người dùng quá nhiều ở một thời điểm. Thiết kế trên Distributed NoSQL giảm thiểu tối đa các phép tính toán, I/O liên quan kết hợp với batch processing đủ đảm bảo được yêu cầu xử lý dữ liệu của các mạng dịch vụ dữ liệu cộng đồng này. Facebook, Amazon là những ví dụ điểm hình.
- b. NoSQL chủ yếu lựa chọn mô hình lưu trữ tập dữ liệu theo cặp giá trị key/value để dễ dàng trong việc truy xuất. Trong NoSQL có khái niệm gọi là node được sử dụng trong quản lý dữ liệu phân tán. Hệ thống dữ liệu phân tán không có sự kiểm tra có trùng dữ liệu hay không, nên dữ liệu phân tá sẽ chấp nhận việc lưu trữ trùng lặp dữ liệu. Khi có một yêu cầu truy vấn lấy dữ liêu có thể có nhiều máy cùng lúc gửi đến, nếu có một máy nào bị hỏng thì cũng không ảnh hưởng nhiều tới toàn bộ hệ thống. Thông thường người ta sẽ tách biệt database ra nhiều database để đảm bảo lượng truy cập xử lý lớn. Một database nhỏ đảm bảo được dữ liệu nhập/ xuất liên tục, nhưng dữ liệu đạt tới ngưỡng nào đó thì sẽ gộp vào một Database lớn hơn có thiết kế tối ưu cho việc. Mô hình đó cho phép tăng cường hiệu suất nhập/ xuất.
- c. Thế hệ CSDL NoSQL giảm thiểu tối đa các phép tính toán, tác vụ đọc-ghi, đảm bảo được yêu cầu xử lý dữ liệu của các dịch vụ mạng xã hội. Hệ CSDL này có thể lưu trữ, xử lý từ lượng rất nhỏ đến hàng petabytes dữ liệu với khả năng chịu tải, chịu lỗi cao nhưng chỉ đòi hỏi về tài nguyên phần cứng thấp.
- d. NoSQL sẽ là lựa chọn tốt nhất cho các công ty nhỏ không đủ kinh phí cho việc đầu tư về bản quyền hoặc server mạnh, cấu hình cao.
- e. Đây là kết quả thống kê về lượng người dùng sử dụng cơ sở dữ liệu NoSQL và RDBMs:

	Rank				Score
May 2015	Apr 2015	May 2014	DBMS	Database Model	May Apr May 2015 2015 2014
1.	1.	1.	Oracle	Relational DBMS	1442.10 -4.03 -60.64
2.	2.	2.	MySQL	Relational DBMS	1294.26 +9.68 -14.83
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1131.03 -18.09 -76.77
4.	4.	↑ 5.	MongoDB 🛅	Document store	277.32 -1.27 +52.70
5.	5.	4 .	PostgreSQL	Relational DBMS	273.52 +5.20 +32.88
6.	6.	6.	DB2	Relational DBMS	201.04 +3.40 +14.57
7.	7.	7.	Microsoft Access	Relational DBMS	145.58 +3.39 +0.22
8.	8.	1 9.	Cassandra 🖰	Wide column store	106.55 +1.66 +24.82
9.	9.	4 8.	SQLite	Relational DBMS	105.16 +2.86 +15.87
10.	10.	1 3.	Redis	Key-value store	94.73 +0.17 +32.69

f. Một so sánh nhỏ về thời gian cập nhật dữ liệu:



2.4. Phân loại NoSQL

- a. Wide Column Store / Column Families: Hệ cơ sở dữ liệu cho phép truy xuất ngẫu nhiên/tức thời với khả năng lưu trức một lượng lớn dữ liệu có cấu trúc. Dữ liệu có thể tồn tại dạng bảng với hàng tỷ bản ghi và mỗi bản ghi có thể chứa hàng triệu cột. Hệ thống triển khai từ vài trăm cho đến hàng nghìn thiết bị dẫn đến khả năng lưu trữ hàng petabytes nhưng vẫn đảm bảo hiệu năng cao.
 - i. Một số sản phẩm thông dụng: Hadoop/Hbase Apache, Bigtable Google, Cassandra Facebook, Hypertable Baidu, Accumulo, Amazon SimpleDB, Cloundata, Clouera...

- ii. Thường cho: các hệ phân tán file
- iii. Ưu điểm: Tìm kiếm nhanh, Phân tán dữ liệu tốt
- iv. Nhược điểm: Hỗ trợ được với rất ít phần mềm
- b. Document Store: Thực chất là các cơ sở dữ liệu hướng tài liệu, một thiết kế riêng biệt cho việc lưu trữ tài liệu. Các cài đặt có thể là giả lập tương tác trên các cở sở dữ liệu quan hệ, cơ sở dữ liệu đối tượng hay key-value store.
 - i. Một số sản phẩm tiêu biểu: MongoDB, Elasticsearch, Couchbase Server, CouchDB, RethinkDB, .
 - ii. Thường cho: Web applications
 - iii. Ưu điểm: Dùng khi dữ liệu nguồn không được mô tả đầy đủ
 - iv. Nhược điểm: Hiệu năng truy vấn, Không có cú pháp chuẩn cho câu truy vấn dữ liệu
- c. Key Value / Tuple Store: Mô hình lưu trữ dữ liệu dưới dạng cặp giá trị key-value trong đó việc truy suất, xóa, cập nhật giá trị thực thông qua key tương ứng. Với sự hỗ trợ của các kĩ thuật Btree, B+Tree, Hash,... dữ liệu có thể tồn tại trên RAM hoặc ổ cứng, phân tán hoặc không phân tán. Hầu hết cá NoSql Database đều là key-value store.
 - i. Các sản phẩm thông dụng: DynamoDB, Azure Table Storage, Riak, Redis,...
 - ii. Thường cho: Content caching Applications
 - iii. Ưu điểm: Tìm kiếm rất nhanh
 - iv. Nhược điểm: Lưu dữ liệu không theo khuôn dạng (schema) nhất định
- d. Graph Databases: Là một dạng cơ sở dữ liệu được thiết kế riêng cho việc lưu trữ thông tin đồ họa như cạnh, nút hay thuộc tính.
 - i. Một số sản phẩm tiêu biểu như:Neo4j, Infinite Graph, InfoGrid, HyperGraphDB, Dex, GraphBase, ...
 - ii. Thường cho: Social networking, Hệ trợ giúp
 - iii. Ưu điểm: Ứng dụng các thuật toán trên đồ thị như Đường đi ngắn nhất, liên thông,...
 - iv. Nhược điểm: Phải duyệt nội bộ đồ thị, để trả lời lại các truy vấn. Không dễ để phân tán
- e. Multimodel Databases: Datomic, OrentDB, ArangoDB, FatDB, AlchemyDB, ...
- f. Object Databases: Versant, db4o, Objectivity, Starcounter, Perst, VelocityDB, HSS Database, ZoDB, ...
- g. Grid & Cloud Database Solutions: Gigaspaces, Infinispan, Queplix, Hazelcast, ...
- h. XML Databases: EMC Document xDB, eXist, Sedna, BaseX, Qizx, ...

ĐỂ TÀI: TÌM HIỆU MONGODE

CHƯƠNG 3. HÊ QUẢN TRI CƠ SỞ DỮ LIÊU MONGODB

- 1. Cơ sở dữ liệu MongoDB
 - 1.1. Giới thiệu MongoDB

MongoDB là một mã nguồn mở, dùng đê thiết kế cơ sở dữ liệu để dễ dàng phát triển và mở rộng, cung cấp hiệu suất cao, tính sẵn sàng cao.

Cấu trúc của Mongodb bao gồm các cặp "thuộc tính" – "giá trị", MongoDB document tương tự như đối tượng JSON. Giá trị của thuộc tính bao gồm các thuộc tính giá trị cụ thể (số, chữ, ...), document, mảng, mảng các đối tượng.

```
"MaCongViec": CV11,
"TenCongViec": aaa,
"MucDo": 1,
"NgayBatDau": 22/1/1993 12:12,
"NgayKetThuc": 1/1/1 12:13,
"NguoiTao": NV1,
"TrangThai": ket thuc_dang lam_ chua bd,
"GhiChu": gc
"NhanVienThucHien": data
            "MaNhanVien": NV1,
            "Quyen": Quan ly, 
"MucDo": 1,
            "GhiChu": fff,
            "TiTrong": 1
            "MaNhanVien": NV2,
             "Quyen": Nhan vien,
            "MucDo": 1,
            "GhiChu": fff,
             "TiTrong": 1
"ConViecCon": {...}
```

Các tính năng chính

- Hiệu suất cao: Hỗ trợ nhúng dữ liệu dạng mô hinh dữ liệu giúp giảm thiểu hoạt động của server, Truy vấn dữ liệu sử dụng chỉ mục giúp tối ưu tốc độ truy vấn.
- Dễ dàng tăng tính mở rộng: chế sharding tự động (tự động phân vùng dữ liệu trên máy chủ, động bộ hóa dữ liệu tốt

MongoDB chạy trên hầu hết các nền tảng và hỗ trợ kiến trúc 64-bit để sử dụng sản xuất và cả hai 64-bit và 32-bit kiến trúc để thử nghiệm.

Các hệ điều hành hổ trợ

	Mongodb	Mongodb Enterprise
Amazon Linux	Hỗ trợ	Hỗ trợ
Debian 7.1	Hỗ trợ	Hỗ trợ
RedHat / CentOS 6.2+	Hỗ trợ	Hỗ trợ
SUSE 11	Hỗ trợ	Hỗ trợ
Ubuntu LTS 12.04	Hỗ trợ	Hỗ trợ
Ubuntu LTS 14.04	Hỗ trợ	Hỗ trợ
Windows Server 2012	Hỗ trợ	Hỗ trợ
Mac OSX 10.6+	Hỗ trợ	
RedHat / CentOS 5.5+	Hỗ trợ	
RedHat / CentOS 5.7+	Hỗ trợ	Hỗ trợ
Windows Server 2008 R2	Hỗ trợ	Hỗ trợ

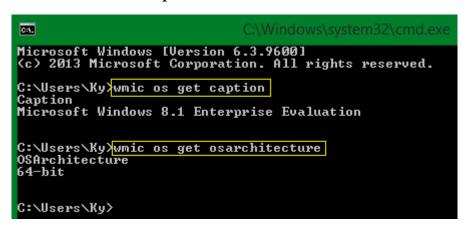
1.2. Cài đặt MongoDB

1.2.1. Giới thiệu về MongoDB trên Windows.

Hướng dẫn cài đặt MongoDB trên Windows, sử dụng hệ điều hành Win8 - 64bit. Và sử dụng phiên bản mongodb: *mongodb-win32-x86_64-2008plus-ssl-3.0.2-signed.msi*

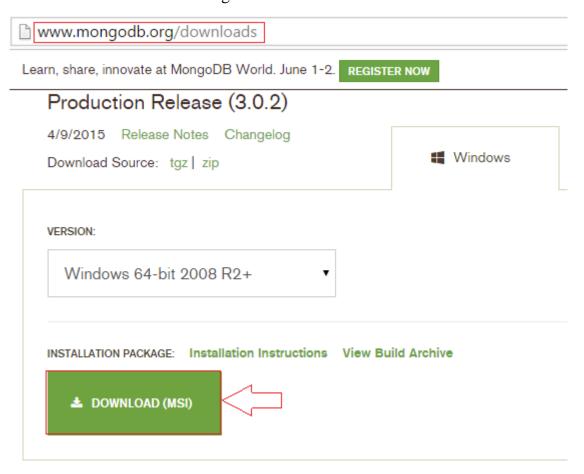
- 1.2.2. Một số chú ý khi cài đặt MongoDB trên Windows
 - 1.2.2.1. Bắt đầu từ phiên bản 2.2, MongoDB không hỗ trợ cho hệ điều hành Windows XP. Vì vậy, nên dùng các bản hệ điều hành mới hơn của Windows.
 - 1.2.2.2. Cài đặt file **.msi** sẽ chứa các phần mềm hỗ trợ tốt cho hệ điều hành đang dùng và tự động cập nhật bản mới cho MongoDB
 - 1.2.2.3. Xác định rõ việc cài đặt loại MongoDB nào cần cho hệ điều hành của bạn:
 - 1.2.2.3.1. MongoDB dành cho Windows 64-bit chỉ chạy trên Windows Server 2008 R2, Windows 7 64-bit và những phiên bản mới hơn của Windows. Phiên bản này tận dụng các cải tiến gần đây cho nền tảng Windows và không hỗ trợ cho các phiên bản trước của Windows.
 - 1.2.2.3.2. MongoDB dành cho Windows 32-bit chỉ chạy trên những hệ điều hành Windows 32-bit mới hơn so với Windows Vista. Phiên bản 32-bit chỉ dành cho những hệ thống cũ và để sử dụng trong thử nghiệm và phát triển hệ thống. Phiên bản 32-bit chỉ hỗ trợ cơ sở dữ liêu nhỏ hơn 2G.
 - 1.2.2.3.3. MongoDB dành cho Windows 64-bit Legacy chạy trên Windows Vista, Windows Server 2003 và Windows Server 2008. Phiên bản này thì không gồm những hỗ trợ tốt nhất về hiệu xuất cũng như nền tảng.

Lhú ý: có thể kiểm tra hệ điều hạnh bạn đang chạy thông qua lệnh "wmic os get caption" và "wmic os get osarchitecture" dùng trong Command Prompt:



1.2.3. Hướng dẫn tải về Windows

Vào trang http://www.mongodb.org/downloads/ kéo xuống dưới và làm theo hướng dẫn như hình:

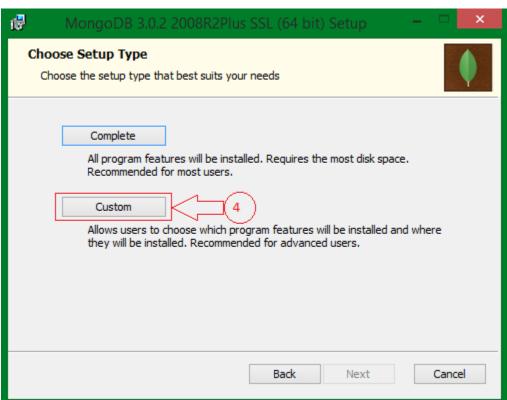


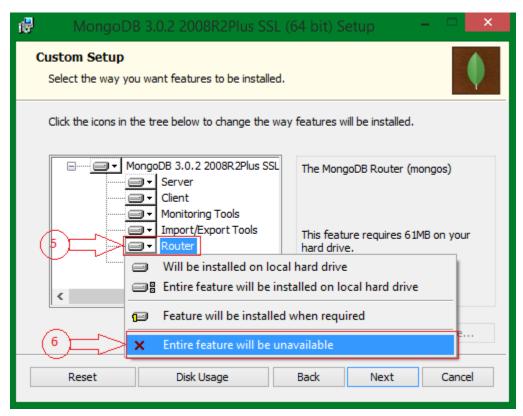
1.2.4. Hướng dẫn cài đặt trên Windows.

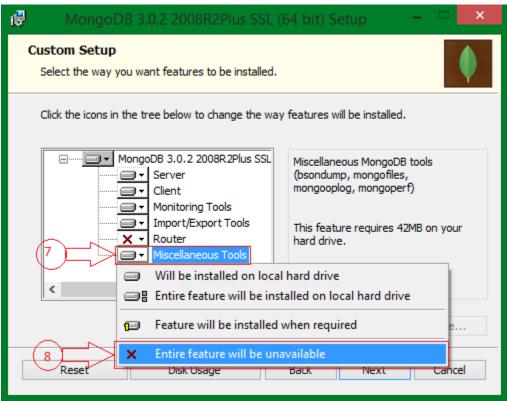
Click đúp vào file mới tải về và làm như sau:

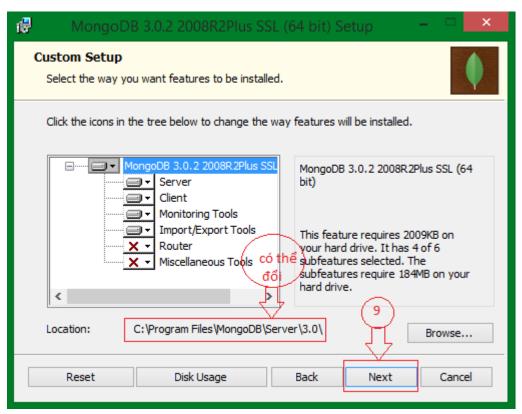


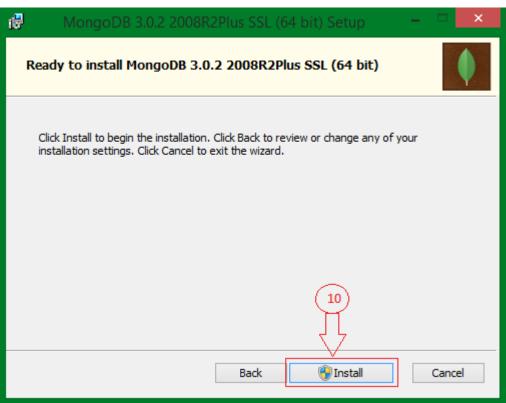






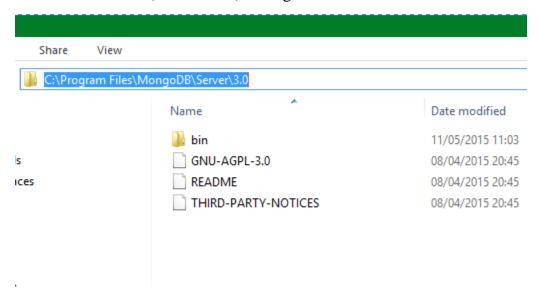




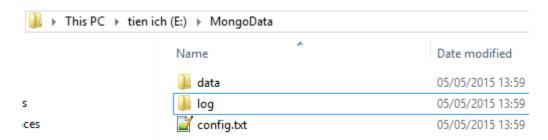




1.2.5. Cấu hình MongoDB trên Windows. Thư mục vừa cài đặt MongoDB:



Bây giờ tạo thư mục cấu hình để lưu dữ liệu, trong thư mục có file log, file cấu hình(Tạo thư mục "MongoData" để chứa dữ liệu, thư mục này gôm có: thư mục data, log và tại 1 file config.txt). Để tránh mất dữ liệu thì nên tránh ổ C ra, tạo các file và thư mục như hình:



Trong file "config.txt" có nội dung như sau:

##store data

 $dbpath=E:\MongoData\data$

##all output go here

logpath=E:\MongoData\log\mongo.log

- Còn 2 thư mục còn lại thì để trống.

Vào **cmd** gõ lệnh: "cd C:\Program Files\MongoDB\Server\3.0\bin" để trỏ đến thư mục cài đặt MongoDB.

```
C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Ky\cd C:\Program Files\MongoDB\Server\3.0
```

Chạy file cấu hình bằng lệnh: "mongod.exe --config E:\MongoData\config.txt"



Giữ nguyên màn hình.

1.2.6. Bắt đầu làm việc với MongoDB

Tao cửa sổ cmd mới và thực hiện như sau:

```
C:\Windows\system32\cmd.exe - mong
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

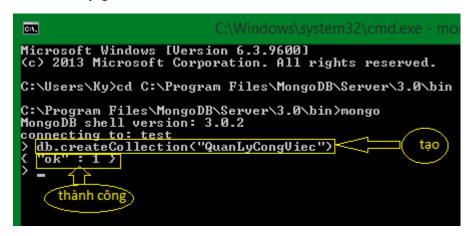
C:\Users\Ky\cd C:\Program Files\MongoDB\Server\3.0\bin\nongo

C:\Program Files\MongoDB\Server\3.0\bin\nongo

MongoDB shell version: 3.0.2
connecting to: test

}
```

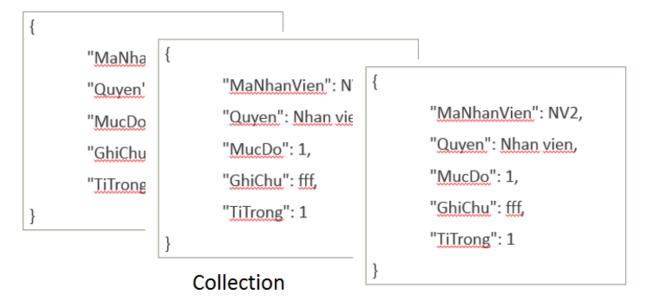
Bây giờ bắt đầu thực hiện các câu lệnh tạo Collection:



- 1.2.7. Công cụ RoboMongo làm việc với MongoDB
 - 1.2.7.1. Giới thiệu: RoboMongo là một công cụ trực quan giúp bạn quản lý Database MongoDB. Nó là một phần mềm mã nguồn mở miễn phí, hỗ trợ cả 3 loại hệ điều hành Windows, Linux, Mac OS.
 - 1.2.7.2. Link tåi: http://robomongo.org/download.html
- 1.3. MongoDB CRUD
 - 1.3.1. Giới thiệu MongoDB CRUD

CRUD là viết tắt của Create (Tạo), Retrieve (Lấy), Update (Cập nhật) và Delete(Xóa). MongoDB lưu trữ dữ liệu trong các document, với các cặp thuộc tính: giá trị như Json.

MongoDB lưu trữ tất cả các document trong Collection. Collection là 1 nhóm các document giống nhau về chỉ số các thành phần.



1.3.2. Các khái niệm trong MongoDB CRUD

♣ Database là gì?

Database là một Ô chứa dữ liệu ở mức vật lý (physical), mỗi database sẽ có nhiều collection và được thiết lập lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDB thường có thể tạo nhiều cơ sở dữ liệu.

♣ Collection là gì?

Collection trong MongoDB là nhóm các tài liệu (document), nó tương đương với một bảng (table) trong CSDL thông thường nên mỗi collection sẽ thuộc về một database duy nhất. Tuy nhiên nó có một sực khác biệt đó là nó không có ràng buộc Relationship như các hệ quản trị CSDL khác nên việc truy xuất rất

nhanh, chính vì thế mỗi collection có thể chứa nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định.

♣ Document là gì?

Document trong MongoDB có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp (key => giá trị) nên nó có tính năng động rất lớn. Document ta có thể hiểu nó giống như các record dữ liệu trong MYSQL, tuy nhiên nó có sự khác biệt là các cặp (key => value) có thể không giống nhau ở mỗi document.

Để rõ hơn chúng ta sẽ so sánh MongoDB và một hệ quản trị CSDL.

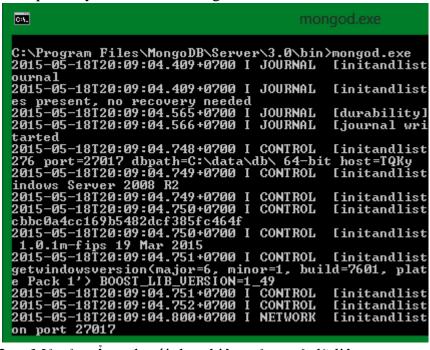
RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Table Join	Embedded Document
Primary key	Primary key (mặc định là _id)

1.3.3. Hướng dẫn MongoDB CRUD

1.3.3.1. Chạy Server: Vào thư mục cài đặt mongo chạy file "mongod.exe" như hình.



Kết quả chạy server thành công.



1.3.3.2. Mở cửa sổ cmd mới thực hiện mở cơ sở dữ liêu:



Kết quả thành công:

```
C:\Program Files\MongoDB\Server\3.0\bin>mongo.exe
MongoDB shell version: 3.0.2
connecting to: test
>
```

1.3.3.3. Tao Database

```
> use QuanLyCongViec tạo Database switched to db QuanLyCongViec kết quả
```

1.3.3.4. Xóa Database

Dùng lênh "show dbs" để xem các database có trong server

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe

MongoDB shell version: 3.0.2
connecting to: test
> show dbs
QuanLyCongViec 0.078GB
abcam 0.078GB
blog 0.078GB
local 0.078GB
test 0.078GB
```

Dùng lệnh "dropDatabase" để xóa 1 database như sau

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe

MongoDB shell version: 3.0.2

connecting to: test
> show dbs

QuanLyCongViec 0.078GB
abcam 0.078GB
blog 0.078GB
local 0.078GB
test 0.078GB
> use QuanLyCongViec
> witched to db QuanLyCongViec
> db.dropDatabase()
{ "dropped" : "QuanLyCongViec", "ok" : 1 }
>
```

Kiểm tra lại các database trong server bằng lệnh "show dbs"

```
C:\Program Files\MongoDB\Server\3.0\bin\mongo.exe

MongoDB shell version: 3.0.2

connecting to: test
> show dbs
QuanLyCongViec 0.078GB
abcam 0.078GB
blog 0.078GB
local 0.078GB
test 0.078GB
> use QuanLyCongViec
switched to db QuanLyCongViec
> db.dropDatabase()
< "dropped": "QuanLyCongViec", "ok": 1 >
> show dbs
abcam 0.078GB
blog 0.078GB
blog 0.078GB
local 0.078GB
test 0.078GB
```

1.3.3.5. Tao Collection

- 1.3.3.6. Thêm dữ liệu
 - 1.3.3.6.1. Thêm dữ liệu chỉ có key/value

```
mongo.exe

> db.Nhom.insert(<_id:12, TenNhanVien: "Tran Trung Hieu">> thêm dữ liệu
WriteResult(< "nInserted" : 1 >> kết quả trả về
>
```

1.3.3.6.2. Thêm dữ liệu có Document

```
mongo.exe

> db.Nhom.insert(<_id:14, NhanVien:(MaNhanVien:1, TenNhanVien:"abc">>>>
WriteResult(< "nInserted": 1 >> kết quả thành công
```

1.3.3.6.3. Thêm dữ liệu có Collection

```
mongo.exe — — X

> db.Nhom.insert(<_id:12, NguoiTao:"Truong Quoc Ky", TenNhom:"Dragon", NgayTao:"
6/6/1993 06:00", ThanhVien:[{MaNhanVien: 1, Quyen: 1}, {MaNhanVien: 2, Quyen: 2}
1>> thêm dữ liệu vào Collection Nhom
WriteResult(< "nInserted": 1 >> kết quả thành công
>
```

- 1.3.3.7. Sửa dữ liệu
 - 1.3.3.7.1. Sửa dữ liêu chỉ có key/value

```
mongo.exe — — — db.Nhom.update</_id:12>,<$set:<NguoiTao:"Tran Trung Hieu">>>\câp nhật dữ liệu iteResult

iteResult
"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >> thành công
```

1.3.3.7.2. Sửa dữ liệu có Document

1.3.3.7.3. Sửa dữ liêu có Collection

1.3.3.8. Xóa dữ liệu

1.3.3.9. Xem dữ liệu của Collection Nhom

```
db.Nhom.find() xem tất cả dữ liệu của Collection Nhom

"_id": 1 }

"_id": 2 }

"_id": 3 }

"_id": 4 }

"_id": 6, "NhanUien": 3 }

"_id": 7, "NhanUien": "3sadf" }

"_id": 5, "NhanUien": "3sadf" }

"_id": 8, "NhanUien": ("MaNhanUien": 1, "TenNhanUien": "TruongQuocKy"

"_id": 10, "NhanUien": [ { "MaNhanUien": 1, "TenNhanUien": "TruongQuoc ] }

"_id": 11, "NhanUien": [ { "MaNhanUien": 1, "TenNhanUien": "TruongQuoc ] }

"_id": 12, "NhanUien": [ { "MaNhanUien": 1, "TenNhanUien": "TruongQuoc | TruongQuoc | TruongQ
```

1.3.3.10. Tìm dữ liệu

```
mongo.exe

> db.Nhom.find<{NhanVien:3>> tim kiếm
{ "_id": 6, "NhanVien": 3 > kết quả trả về
```

1.3.3.11. Xem tất cả các Collection đã tạo



1.3.3.12. Lấy tất cả và sắp xếp theo _id

1.3.3.13. Tìm kiếm gần đúng

```
mongo.exe

> db.Nhom.find({NhanVien:/3/}) tìm kiếm gần đúng
("_id": 7, "NhanVien": "3sadf")
("_id": 5, "NhanVien": "3sadf")

**Kết quả trả về
```

1.3.3.14. df

1.4. Data Model

1.4.1. Giới thiêu Data Model.

Hệ quản trị cơ sở dữ liệu MongoDB là một mô hình dữ liệu linh hoạt. Không giống như các cơ sở dữ liệu SQL, phải xác định và triển khai sơ đồ của một bảng trước khi chèn dữ liệu, Collection trong MongoDB rất linh hoạt, không nhất thiết phải cho biết những thuộc tính có trong Collection, khi nào cần bổ sung sau vẫn được. Sự linh hoạt này tạo điều kiện cho việc hướng tới một cách lưu trữ như một thực thể hay một đối tượng. Mỗi Document có phù hợp với các trường dữ liệu của Collection chứa nó, ngay cả khi thay đổi gần như toàn bộ dữ liệu. Tuy nhiên, trong thực tế, các Document trong một Collection chung một cấu trúc.

Khi thiết kế cơ sở dữ liệu cần chú ý về cân bằng nhu cầu của các ứng dụng (tức là có nên tạo nhiều Collection lồng nhau), các đặc tính hiệu suất khi thực hiện truy vấn, và các mô hình dữ liệu. Khi thiết kế các mô hình dữ liệu, luôn luôn xem xét việc sử dụng các dữ liệu (ví dụ như truy vấn, cập nhật, và xử lý các dữ liệu) cũng như các cấu trúc vốn có của bản thân dữ liệu.

- 1.4.2. Các khái niệm cơ bản về Data Model.
 - 1.4.2.1. Thiết kế Data Model.
 - 1.4.2.1.1. Data Model nhúng.

Với MongoDB, bạn có thể nhúng các dữ liệu liên quan vào một Structure hoặc Document, hay còn được gọi là mô hình "denormalized". Đây là ví dụ:

```
Tài Khoản
                 "MaNhanVien": NV1,
                 "TenNhanVien": aa,
                 "Quyen": Q1,
                 "TrangThai": nghi phep,
                 "TaiKhoan":
                                  "TenTaiKhoan": tk1,
                                  "MatKhau": ***,
                                  "MatKhauCap1": 123,
                                  "MatKhauCap2": 333,
                                  "Mail": a@gmail.com
                 "GhiChu":
                 "SoLuongBanBe": 1,
                 "BanBe": data
                                 {
                                          "MaNhanVien": NV2,
                                          "NgayKetBan": 23,
                                          "MucDo": 2
                                 },...
                         ]
        }
```

Mô hình dữ liệu nhúng cho phép các ứng dụng lưu trữ phần dữ liệu liên quan của thông tin trong Document của cơ sở dữ liệu.

Sử dụng nhúng mô hình dữ liệu khi: Cơ sở dữ liệu có muối quan hệ giữa các thực thể (One-to-One và One-to-Many).

Nói chung, nhúng cung cấp hiệu suất tốt hơn cho hoạt động đọc, cũng như khả năng yêu cầu và lấy dữ liệu có liên quan trong một hoạt động cơ sở dữ liệu duy nhất.

1.4.2.1.2. Chuẩn hóa Data Model.

Tạo liên kết giữa các "id" của các Document với nhau.

```
Quyen
       {
             "MoTa": aaa,
                "MucDo": 1,
                "LoaiQuyen": Nhom NV
        }
Tài Khoản
                " id": NV1,
               "TenNhanVien": aa,
               "Quyen id": <ObjectId1>,
                "TrangThai": nghi phep,
               "TaiKhoan":
                               "TenTaiKhoan": tk1,
                                "MatKhau": ***,
                               "MatKhauCap1": 123,
                               "MatKhauCap2": 333,
                               "Mail": a@gmail.com
                "GhiChu": b,
               "SoLuongBanBe": 1,
               "BanBe": data
                               {
                                        "MaNhanVien": NV2,
                                        "NgayKetBan": 23,
                                        "MucDo": 2
                               },...
                        ]
       }
```

1.4.2.2. Operational Factors and Data Models.

Mô hình hóa dữ liệu ứng dụng cho MongoDB phụ thuộc vào dữ liệu, cũng như các đặc điểm của MongoDB.

1.4.2.2.1. Document Growth.

Một số cập nhật các Document có thể tăng kích thước của văn bản. Những cập nhật này bao gồm đẩy yếu tố để một mảng (tức là \$ push) và thêm các trường mới vào một tài liệu.

Khi sử dụng công cụ lưu trữ MMAPv1, Document Growth có thể được xem xét cho một Data Model của bạn. Đối với MMAPv1,

nếu kích thước dữ liệu vượt quá không gian được chia cho Document đó, MongoDB sẽ di dời các Document trên ổ đĩa.

1.4.2.2.2. Indexes.

Sử dụng chỉ số để cải thiện hiệu suất cho các truy vấn phổ biến. Xây dựng các chỉ số trên các thuộc tính mà xuất hiện thường xuyên trong các truy vấn và cho tất cả các yêu cầu đó trả lại kết quả được sắp xếp. MongoDB tự động tạo ra một chỉ số duy nhất trên các thuộc tính là _id.

Khi tạo Indexes cần chú ý:

- Mỗi Indexes yêu cầu tối thiểu 8KB dữ liệu lưu trữ.
- Thêm một Indexes có một số tác động tiêu cực cho các hoạt động ghi. Vì mỗi khi thêm dữ liệu phải cập nhật chỉ số.
- Càn chú ý đến việc thiết lập kích thước cho từng những thuộc tính vì Indexes sẽ chiếm không gian ổ đĩa và ô nhớ.
- 1.4.3. Data Model Examples and Patterns.
 - 1.4.3.1. Mô hình qua hệ giữa các Document.
 - 1.4.3.1.1. Mô hình One-to-One quan hệ với nhúng Document.
 - 1.4.3.1.2. Mô hình One-to-Many quan hệ với nhúng Document.
 - 1.4.3.1.3. Mô hình One-to-Many quan hệ với tham chiếu giữa các Document.
 - 1.4.3.2. Mô hình cấu trúc cây.
 - 1.4.3.2.1. Mô hình cấu trúc cây với tham chiếu Parent.
 - 1.4.3.2.2. Mô hình cấu trúc cây với tham chiếu Child.
- 1.5. Adminitration
- 1.6. Security
- 1.7. Aggregation
- 1.8. Indexes
- 1.9. Replication

Giống như hệ quản trị CSDL quan hệ, MongoDB hỗ trợ replication của database theo thời gian thực hoặc gần với thời gian thực. Replication trong MongoDB rất đơn giản để cài đặt và sử dụng. Đó cũng là một trong những điểm đặc biệt của hệ quản trị cơ sở dữ liệu thế hệ mới, thế hệ cho nền tảng web 2.0 và công nghệ lưu trữ điện toán đám mây. Có nhiều kịch bản đặt ra cho việc sử dụng Replication. MongoDB hỗ trợ đầy đủ các trường hợp Replication mà thực tế đặt ra.

Hiện nay MongoDB hỗ trợ các kịch bản Replication sau:

• Single master/single slave replication

- Single master/multiple slave replication
- Multiple master/single slave replication
- Cascade replication
- Master/master replication
- Interleaved replication
- Replica pairs (legacy MongoDB clustering)
- Replica sets (advanced MongoDB clustering)

1. Muc đích Replication của MongoDB

Replication đem lại nhiều lợi ích tùy thuộc và loại Replication mà bạn chọn và những tùy chọn mà chúng ta chọn. Nhưng một trong những lợi ích thông thường của Replication là dữ liệu luôn sẵn sàng mọi lúc mọi nơi khi chúng ta cần.

2. Khi nào thì sử dụng Replication

Replication là giải pháp được ứng dụng cho môi trường phân phối dữ liệu trên nhiều Server, chính vì vậy mà sử dụng chúng khi:

- o Sao chép và phân phối dữ liệu trên nhiều Server khác nhau
- o Phân phối bản sao dữ liệu theo lịch trình nhất định.
- o Phân phối dữ liệu vừa thay đổi trên nhiều Server khác nhau.
- o Cho phép nhiều người dùng và nhiều Server kết hợp dữ liệukhacs nhau một cách thống nhất mà không sơ mất dữ liêu.
- o Xây dựng CSDL sử dụng cho những ứng dụng trực tuyến hay ngoại tuyến
- o Xây dụng ứng dụng Web khi người dùng cần trình bày một số lượng lớn dữ liệu.

1.10. Sharding

2. Một số đặc trung của Mongodb

Thuộc tính	Đặc trưng
Lưu trữ hướng văn bản	Văn bản theo phong cách JSON với
	những lược đồ động đơn giản
Hỗ trợ chỉ mục đầy đủ	Chỉ mục trên bất kỳ các thuộc tính.
Tính sao lặp và tính sẵn sàng cao	Mở rộng.
Auto-sharding	Mở rộng theo chiều ngang mà không ảnh hưởng đến chức năng
Truy vấn	Đa dạng, truy vấn dựa trên văn bản
GridFS	Lưu trữ file với bất kỳ kích cỡ nào mà không làm phức tạp ngăn xếp

Hỗ trợ thương mại	Hỗ trợ doanh nghiệp, đào tào, tư vấnKhái niệm cơ bản trong Mongodb

- Các điểm lưu ý khi lựa chọn MongoDB:
 - Nếu ứng dụng của có tính chất INSERT cao, bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.
 - Úng dụng ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dung. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì nó sẽ bỏ qua phần đó nên sẽ an toàn.
 - Úng dụng bạn có nhiều dữ liệu quá. Ví dụ, giả sử web có đến 10 triệu records thì đó là cơn ác mộng với MYSQL. Bởi vì MongoDB có khả năng tìm kiến thông tin liên quan cũng khá nhanh nên trường hợp này nên dùng nó.
 - Máy chủ không có hệ quản trị CSDL, trường hợp này thường bạn sẽ sử dụng SQL LIFE hoặc là MongoDB.
- 3. Sử dụng cơ bản một số lệnh trong MongoDB
- 4. Một số lệnh truy vấn tương đương giữa SQL và MongoDB

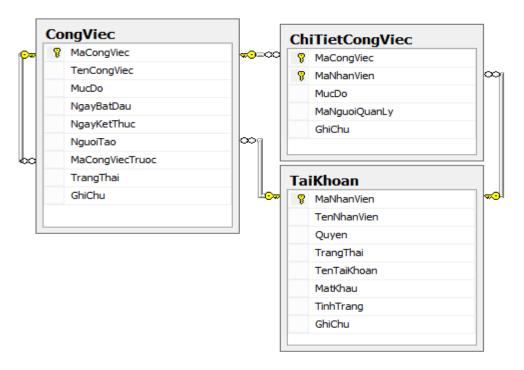
Câu lệnh SQL SERVER	Câu lệnh Mongo
CREATE TABLE USERS (a Number, b Number)	db.createCollection("mycoll")
INSERT INTO USERS VALUES(3,5)	db.users.insert({a:3,b:5})
SELECT a,b FROM users	db.users.find({}, {a:1,b:1})
SELECT * FROM users	db.users.find()
SELECT * FROM users WHERE age=33	db.users.find({age:33})
SELECT a,b FROM users WHERE age=33	db.users.find({age:33}, {a:1,b:1})
SELECT * FROM users WHERE age=33 ORDER BY name	db.users.find({age:33}).sort({name:1})
SELECT * FROM users WHERE age>33	db.users.find({age:{\$gt:33}})
SELECT * FROM users WHERE age!=33	db.users.find({age:{\$ne:33}})
SELECT * FROM users WHERE name LIKE "%Joe%"	db.users.find({name:/Joe/})
SELECT * FROM users WHERE name LIKE "Joe%"	db.users.find({name:/^Joe/})
SELECT * FROM users WHERE age>33 AND age<=40	db.users.find({'age':{\$gt:33,\$lte:40}})
SELECT * FROM users ORDER BY name DESC	db.users.find().sort({name:-1})
SELECT * FROM users WHERE a=1 and b='q'	db.users.find({a:1,b:'q'})
SELECT * FROM users LIMIT 10 SKIP 20	db.users.find().limit(10).skip(20)

ĐỂ TÀI: TÌM HIỂU MONGODB

SELECT * FROM users WHERE a=1 or b=2	db.users.find({ \$or : [{a:1} , {b:2}] })
SELECT * FROM users LIMIT 1	db.users.findOne()
SELECT order_id FROM orders o, order_line_items li WHERE li.order_id=o.order_id AND li.sku=12345	db.orders.find({"items.sku":12345},{_id:1})
SELECT customer.name FROM customers,orders WHERE orders.id = "q179"	var o = db.orders.findOne({_id:"q179"});
AND orders.custid = customer.id	var name = db.customers.findOne({_id:o.custid})
SELECT DISTINCT last_name FROM users	db.users.distinct('last_name')
SELECT COUNT(*y)	db.users.count()
FROM users	
SELECT COUNT(*y)	db.users.find({age: {'\$gt': 30}}).count()
FROM users where AGE > 30	
SELECT COUNT(AGE) from users	db.users.find({age: {'\$exists': true}}).count()
CREATE INDEX myindexname ON users(name)	db.users.ensureIndex({name:1})
CREATE INDEX myindexname ON users(name,ts DESC)	db.users.ensureIndex({name:1,ts:-1})
EXPLAIN SELECT * FROM users WHERE z=3	db.users.find({z:3}).explain()
UPDATE users SET a=1 WHERE b='q'	db.users.update({b:'q'},{\$set:{a:1}},false,true)
UPDATE users SET a=a+2 WHERE b='q'	db.users.update({b:'q'},{\$inc:{a:2}},false,true)
DELETE FROM users WHERE z="abc"	db.users.remove({z:'abc'});

CHƯƠNG 4. XÂY DỰNG ỨNG DỤNG VỚI MONGODB

- 4.1. Giới thiệu ứng dụng
- 4.2. Thiết kế và tạo CSDL trên Mongodb CSDL

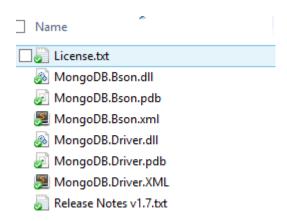


4.3. Xây dựng ứng dụng kết nối Mongodb

4.3.1. Chuẩn bị thư viện kết nối với Mongodb

Hướng dẫn: http://docs.mongodb.org/ecosystem/drivers/csharp/ Download: https://github.com/mongodb/mongo-csharp-driver

Giải nén ta được các file .DLL



4.3.2. Tạo lớp DataService để thao tác với CSDL MogoDB

4.3.2.1. Tao DataBase

Xây dựng lớp DataService gôm có 1 biến MongoDatabase để nhận database khi kết nối được với server

```
public class DataService
{
    MongoDatabase db;
    string ConnectionString = "Server=localhost:27017";

    public DataService()
    {
        MongoServer server = MongoServer.Create(ConnectionString);
        db = server.GetDatabase(DTO.DTO_QuanLyCongViec.NAME_DATABASE);
    }
}
```

4.3.2.2. Thêm dữ liệu cho 1 Collection

Tạo ra lớp đối tượng tương ứng với Collection đó

```
public class DTO_LoaiQuyen
{
    public ObjectId _id { get; set; }
    public BsonInt32 MaLoai { get; set; }
    public string MoTa { get; set; }
    public BsonInt32 MucDo { get; set; }
    public string ChuThich { get; set; }

    public DTO_LoaiQuyen()
    { }
}
```

Dùng đối tượng BsonDocument để thêm vào MongoCollection<BsonDocument>

Thử dữ liệu vào ở tầng giao diện

Vào CMD gõ lênh "db.LoaiQuyen.find()" để xem dữ liệu trong CMD

4.3.2.3. Lấy dữ liêu từ Collection

Lấy tất cả dữ liệu của 1 Collection: Lấy dữ liệu của Collection LoaiQuyen

```
public List<DTO_LoaiQuyen> GetAll()
{
    List<DTO_LoaiQuyen> kq = new List<DTO_LoaiQuyen>();
    var query = new QueryDocument();
    BsonDocument document = new BsonDocument();

    MongoServer server = MongoServer.GetAllServers()[0];
    MongoDatabase dbmg = server.GetDatabase(DTO_QuanLyCongViec.NAME_DATABASE);
    var tmp = dbmg.GetCollection<DTO_LoaiQuyen>(NAME_CLASS);
    foreach (DTO_LoaiQuyen item in tmp.FindAll())
    {
        kq.Add(item);
    }
    return kq;
}
```

4.3.2.4. Xóa 1 Collection

Lấy đúng collection từ trong database theo tên collection sau đó gọi hàm drop của MongoCollection

```
public void DropLoaiQuyen()
{
          MongoServer server = MongoServer.GetAllServers()[0];
          MongoDatabase dbmg = server.GetDatabase(DTO_QuanLyCongViec.NAME_DATABASE);
          MongoCollection<DTO_LoaiQuyen> collection = dbmg.GetCollection<DTO_LoaiQuyen>(DAL_LoaiQuyen.NAME_CLASS collection.Drop();
}
```

Kiểm tra lại dữ liệu trong CMD trước và sau khi gọi hàm.

```
wiched to db QuantyCongViec
vitched to Q
```

CHƯƠNG 5. KẾT LUẬN

TÀI LIỆU THAM KHẢO

- [1]. **Ebook NoSQL** Nhữ Đình Thuận
- [2]. Eelco Plugge, Peter Membrey and Tim Hawkins, The Definitive Guide to MongoDB The NoSQL Database for Cloud and Desktop Computing.
- [3]. **NoSQL resources** http://nosql-database.org/
- [4]. NoSQL in the Enterprise http://www.infoq.com/articles/nosql-in-the-enterprise
- [5]. NoSQL wiki http://en.wikipedia.org/wiki/NoSQL
- [6]. **Kristina Chodorow Michael Dirolf, MongoDB**: The Definitive Guide, O'reilly, 2010.
- [7]. http://www.mongodb.org