**BÁO CÁO NHÓM 5: NOSQL**

**Công việc:**

**- Tạo repository trên github: Đã thực hiện**

**- Tìm hiểu về NOSQL: Đã thực hiện**

**- Chủ Đề: Web bán mặt hàng điện thoại di động**

**- Ngôn ngữ: Java**

**- Hệ quản trị cơ sở dữ liệu: MONGODB**

**- Tìm hiểu về mongoDB: Đã thực hiện**

**- Cài Đặt và cấu hình mongoDB: đã thực hiện**

**TIẾN ĐỘ**

1. **NOSQL**
2. NoSql là gì

* NoSql có nghĩa là Non-Relational
* NoSql là Not only SQL
* không sử dụng mô hình dữ liệu quan hệ (RDBMs)
* Mô hình lưu trữ {“key”:value”}
* Hệ thống lưu trữ phân tán
* Nosql là hệ cơ sở không rang buộc, phân tán, mã nguồn mở, khả năng mở rộng theo chiều ngang, có thể chưa hàng petabytes, độ chịu tải và chịu lỗi cao, yêu cầu về tài nguyên phần cứng thấp.

1. Lịch sử phát triển

* Giới thiệu lần đầu vào năm 1998
* Lightweight open source relational database
* Năm 2009, eric evans, nhân viên của rackspace
* Nosql: phân tán + không ràng buộc
* Nosql storage đặc biệt thông dụng với mạng xã hội

1. một số thuật ngữ

* Ràng buộc (như khóa chính, ngoại, 1-n…)
* Không ràng buộc ( không ràng buộc giữa tập hợp với nhau)
* Khả năng mở rộng (có thể bổ sung thêm dữ liệu)
* Khả năng mở rộng theo chiều dọc ( dùng để tăng bộ nhớ, tăng ram, thay đổi ổ cứng máy chủ…)
* Khả năng mở rộng theo chiều ngang
* Phân tán dữ liệu ( dữ liệu đc lưu trong nhiều sever)
* Triển khai linh hoạt ( thêm hay tắt sever đều ko ảnh hưởng các sever khác)
* Tính sẵn sàng ( nếu 1 sever có vấn đề thì ko ảnh hưởng đến sever khác)
* Nhất quán cuối ( đến cuối cùng mọi thứ sẽ đồng bộ với nhau)
* Lưu trữ tốt

1. So Sánh SQL và NOSQL

|  |  |  |
| --- | --- | --- |
| **Tính năng** | SQL | NOSQL |
| Hiệu xuất | Kém hơn Nosql vì khi truy vấn nó phải tính toán, kiểm tra và xử lý các mối quan hệ. | Tốt SQL vì nó bỏ qua các rang buộc |
| Mở rộng theo chiều ngang | Có thể thực hiện được nhưng quá trình mở rộng sẽ rất phức tạp nếu đã tồn tại dữ liệu trong Database | Mở rộng dễ dàng |
| Tốc độ Read/write | Kém hơn so với nosql vì đảm bảo tính rang buoocjj dữ liệu giữ các bản  Nếu sử dụng nhiều server thì phải bảo toàn tính nhất quán về dữ liệu ở các server với nhau | Tốc độ nhanh hơn SQL vì bỏ qua tính rang buộc của các bản  Vì dữ liệu được lưu trong ram, sau đó đẩy xuống HDD và tính nhất quán cuối |
| Phần cứng | Đòi hỏi phần cứng cao | Không đòi hỏi quá nhiều phần cứng |
| Thay đổi số node trong hệ thống | Vì tính nhất quán về dữ liệu nên khi thêm hay xóa 1 node cần phải shutdown hệ thống trong 1 khoảng thới gian | Vì tính nhất quán không cần phải shutdown hệ thống |
| Truy vấn và báo cáo | Dễ dàng sử dụng ngôn ngữ SQL query để truy vấn trực tiếp dữ liệu từ database hoặc dung công cụ hỗ trợ để lấy báo cáo | Việc lấy báo cáo dữ liệu trực tiếp từ NoSQL chưa được hỗ trỡ tốt, thực hiện chủ yếu thông qua giao diện ứng dụng |
| Mở rộng dữ liệu | Khi muốn mở rộng dữ liệu chúng ta phải khai báo trước | Không cần báo trước |
| Ứng dụng | Sử dụng để xây dựng những hệ thống có quan hệ chặt chẽ và cần tính đông nhất về dữ liệu như: tài chính, ngân hang, chứng khoáng… | Sử dụng xây dựng những hệ thống lưu trữ thông tin lớn, không quá quan trọng về vấn đề đồng nhất dữ liệu trong một khoảng thời gian nhất ddingj. VD như: báo trí, mạng XH, diễn đàng…. |

1. Key-values Stores

Key value stores, hoặc key value database là kiểu cơ sở dữ liệu sử dụng mảng kết hợp. Trong đó mỗi key sẽ tồn tại duy nhất một value trong collection. Mối quan hệ này được gọi là một cặp key-value Examples: Aerospike, CouchDB, Dynamo, FairCom c-treeACE, FoundationDB, HyperDex, MemcacheDB, MUMPS, Oracle NoSQL Database, OrientDB, Redis, Riak, Berkeley DB

1. Column Family Stores

Ý tưởng cơ bản: o Column families: Một column family là cách thức dữ liệu được lưu trữ trên đĩa. Tất cả dữ liệu trong một cột sẽ được lưu trên cùng một file. Examples: Hadoop/HBase – Apache, BigTable – Google, Cassandra - Facebook/Apache, Hypertable - Zvents Inc/Baidu, Cloudera, SciDB, Mnesia, Tablets,…

1. Document Databases

Document stores, hay còn gọi là hệ thống database hướng document (document-oriented) là một kiểu database mà ở đó data được tổ chức một cách tự do không theo một lược đồ nào cả (schema-free).

Điều này có nghĩa là:

* Mỗi bản ghi không cần phải có cấu trúc cố định, các bản ghi khác nhau có thể có nhiều cột khác nhau.
* Loại dữ liệu trong mỗi cột giữa các bản ghi cũng có thể khác nhau
* Một cột có thể có nhiều hơn một giá trị (mảng – array)
* Các bản ghi có thể có cấu trúc lồng nhau (trong một bản ghi chứa một hoặc nhiều bản ghi khác – nested structure
* Examples: Apache Jackrabbit, CouchDB, IBM Lotus Notes Storage Format (NSF), MongoDB, Terrastore, ThruDB, OrientDB, RavenDB,...

1. Graph Databases

Cơ sở dữ liệu đồ thị (graph database) là cơ sở dữ liệu được thiết kế để coi các mối quan hệ giữa các dữ liệu là quan trọng như nhau đối với bản thân dữ liệu. Nó được thiết kế để lưu giữ dữ liệu mà không cần biến nó thành một mô hình được xác định trước. Examples: Neo4J, Sones, AllegroGraph, Core Data, DEX, FlockDB, InfoGrid, OpenLink Virtuoso,...

1. **MONGODB**

1. Định nghĩa về MongoDB

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL(\*) hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++. Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

2. 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 mấy 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.

4. Các kiểu dữ liệu trong MongoDB

MongoDB hỗ trợ các kiểu dữ liệu sau:

| **Type** | **Number** | **Alias** |
| --- | --- | --- |
| Double | 1 | “double” |
| String | 2 | “string” |
| Object | 3 | “object” |
| Array | 4 | “array” |
| Binary data | 5 | “binData” |
| Undefined | 6 | “undefined” |
| ObjectId | 7 | “objectId” |
| Boolean | 8 | “bool” |
| Date | 9 | “date” |
| Null | 10 | “null” |
| Regular Expression | 11 | “regex” |
| DBPointer | 12 | “dbPointer” |
| JavaScript | 13 | “javascript” |
| Symbol | 14 | “symbol” |
| JavaScript (with scope) | 15 | “javascriptWithScope” |
| 32-bit integer | 16 | “int” |
| Timestamp | 17 | “timestamp” |
| 64-bit integer | 18 | “long” |
| Decimal128 | 19 | “decimal” |
| Min key | -1 | “minKey” |
| Max key | 127 | “maxKey” |

**Chuỗi:** Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải là UTF-8 hợp lệ.

**Số nguyên:** Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên có thể là 32 bit hoặc 64 bit phụ thuộc vào Server của bạn.

**Boolean:** Kiểu dữ liệu này được sử dụng để lưu giữ một giá trị Boolean (true/false).

**Double:** Kiểu dữ liệu này được sử dụng để lưu các giá trị số thực dấu chấm động.

**Min/ Max keys:** Kiểu dữ liệu này được sử dụng để so sánh một giá trị với các phần tử BSON thấp nhất và cao nhất.

**Mảng:** Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.

**Timestamp:** Đánh dấu thời điểm một Document được sửa đổi hoặc được thêm vào.

Object: Kiểu dữ liệu này được sử dụng cho các Document được nhúng vào.

**Null:** Kiểu dữ liệu này được sử dụng để lưu một giá trị Null.

**Symbol**: Kiểu dữ liệu này được sử dụng giống như một chuỗi

**Date :** Kiểu dữ liệu này được sử dụng để lưu giữ date và time hiện tại trong định dạng UNIX time.

**Object ID:** Kiểu dữ liệu này được sử dụng để lưu giữ ID của Document.

**Binary data:** Kiểu dữ liệu này được sử dụng để lưu giữ dữ liệu nhị phân.

**Code:** Kiểu dữ liệu này được sử dụng để lưu giữ JavaScrip code vào trong Document.

**Regular expression:** Kiểu dữ liệu này được sử dụng để lưu giữ Regular Expresion.

1. MongoDB hoạt động như thế nào

* MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
* Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.
* Từ đây có thể nhìn thấy nhược điểm của Mongodb như sau:

Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

1. Lợi thế của MongoDB

Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, dễ 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.

Cấu trúc của một đối tượng rõ ràng.

Không có các Join phức tạp.

Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi hơn.

Sử dụng bộ nhớ trong để lưu giữ cửa sổ làm việc cho phép truy cập dữ liệu nhanh hơn. Việc cập nhật được thực hiện nhanh gọn nhờ update tại chỗ (in-place).

1. Khi nào NÊN sử dụng MongoDB ?

Sử dụng MongoDB trong trường hợp:

Nếu website của bạn có tính chất INSERT cao Bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.Website của bạn 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.

Website bạn có nhiều dữ liệu quá Giả sử web bạn 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 SQLITE hoặc là MongoDB.

1. Khi nào KHÔNG NÊN sử dụng MongoDB ?

Các ứng dụng cần sử dụng nhiều transaction (như ngân hàng) do Mongodb không có cơ chế transaction (giao dịch) để phục vụ cho các ứng dụng ngân hàng

Các ứng dụng cần SQL (sử dụng joins).

1. Cài đặt và cấu hình mongoDB

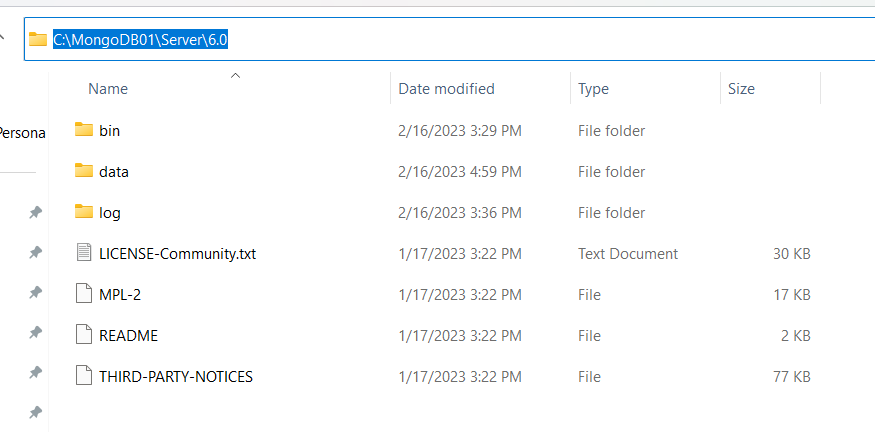
Bước 1: truy cập vào đường link để tải mongoDB: <https://www.mongodb.com/try/download/community>

Bước 2: sau khi tải về bạn tiến hành cài dặt như bình thường

Bước 3: cấu hình

+ vào mục Search trên window gõ lệnh cmp: để mở Command Prompt

+ sau đó bạn tìm đến nơi lưu file mongodb và copy đường dẫn path vào phần cmp



+ sau đó gõ lệnh: mongod để chạy. lúc đấy bạn vẫn chưa chạy được phải cần tạo 1 foder bên ngoài có tên là data\db

Graphical user interface, text, application

Description automatically generated

+ sau đó bạn vào link bên dưới để tải một phần để quản lý database: https://www.mongodb.com/try/download/compass

+ sau khi bạn cài xong thì kích vào để chạy và sẽ được giao diện như bên dưới.

+ link tham khảo: https://www.youtube.com/watch?v=GKHTa6Z07bc&list=PLv6GftO355Aug0rwKfb6v96mlYrwOw7XV&index=9

Graphical user interface, text, application

Description automatically generated