ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO BÀI TẬP NHÓM MÔN HỌC**

CƠ SỞ DỮ LIỆU NÂNG CAO

Chủ đề:

**TÌM HIỂU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU COUCHBASE**

**Giảng viên hướng dẫn**: PGS. TS Nguyễn Ngọc Hóa

**Nhóm học viên** : Nguyễn Tiến Uy

Đỗ Đức Cường

Nguyễn Công Thắng

**Nhóm** : 03

Hà Nội, tháng 1 năm 2019

**MỤC LỤC**

[I. Yêu cầu tìm hiểu 4](#_Toc535600948)

[II. Giới thiệu Couchbase 5](#_Toc535600949)

[III. Mục đích và đối tượng chính của DBMS, 6](#_Toc535600950)

[1. Mục đích 6](#_Toc535600951)

[2. Đối tượng chính 6](#_Toc535600952)

[IV. Mô hình dữ liệu 6](#_Toc535600953)

[1. Dữ liệu 7](#_Toc535600954)

[2. Lưu dữ liệu trong Bucket, Bộ nhớ 8](#_Toc535600955)

[3. Ngôn ngữ truy vấn dữ liệu 8](#_Toc535600956)

[V. Phương pháp kỹ thuật xử lý giao tác đồng thời 10](#_Toc535600957)

[1. Kiến trúc xử lý 10](#_Toc535600958)

[1.1. Mô hình cluster 10](#_Toc535600959)

[1.2. Mô hình của một node: 11](#_Toc535600960)

[1.3. Truy vấn dữ liệu 12](#_Toc535600961)

[VI. Hiệu năng, so sánh và đánh giá 13](#_Toc535600962)

[1. Hiệu suất 13](#_Toc535600963)

[1.1. Import dữ liệu 13](#_Toc535600964)

[1.2. Truy vấn dữ liệu 13](#_Toc535600965)

[1.3. Cập nhật dữ liệu 13](#_Toc535600966)

[1.4. Join bảng 14](#_Toc535600967)

[1.5. Chèn 14](#_Toc535600968)

[2. Đánh giá 14](#_Toc535600969)

[VII. Tài liệu cài đặt và cấu hình 14](#_Toc535600970)

[1. Couchbase 14](#_Toc535600971)

[1.1. Cài đặt: 14](#_Toc535600972)

[1.1. Địa chỉ truy cập server 14](#_Toc535600973)

[1.2. Tạo cluster, cấu hình tài nguyên cho cluster 14](#_Toc535600974)

[1.3. Tạo bucket, cấu hình tài nguyên cho bucket 14](#_Toc535600975)

[1.4. Tạo index 14](#_Toc535600976)

[1.5. Import dữ liệu 15](#_Toc535600977)

[1.6. Tắt/bật dịch vụ server 17](#_Toc535600978)

[2. mysql 17](#_Toc535600979)

[2.1. Cài đặt: 17](#_Toc535600980)

[2.2. Kết nối tới server: 17](#_Toc535600981)

[2.3. Tạo db 17](#_Toc535600982)

[2.4. Sử dụng DB: 17](#_Toc535600983)

[2.5. Kiểm tra DB 17](#_Toc535600984)

[2.6. Tạo bảng 18](#_Toc535600985)

[2.7. Kiểm tra bảng 18](#_Toc535600986)

[2.8. Cấu hình cho phép import dữ liệu local 18](#_Toc535600987)

[2.9. Import dữ liệu vào bảng 18](#_Toc535600988)

[2.10. Tắt dịch vụ/bật dịch vụ mySQL Server 18](#_Toc535600989)

[Tài liệu tham khảo 19](#_Toc535600990)

# Yêu cầu tìm hiểu

Lựa chọn, nghiên cứu về một hệ quản trị CSDL hiện đại với những yêu cầu chính sau:

* Mục đích và đối tượng chính của DBMS,
* Mô hình dữ liệu được sử dụng trong DBMS,
* Phương pháp và kỹ thuật xử lý các truy vấn/giao tác đồng thời,
* So sánh, đánh giá hiệu năng DBMS đó so với một DBMS truyền thống (chẳng hạn như mySQL) thông qua thực nghiệm với ứng dụng trên Web:
  + Quy mô dữ liệu khoảng 100.000.000 bộ/tài liệu
  + Có sử dụng những truy vấn phức tạp (join, phân mảnh, ...)

Kết quả nghiên cứu, tìm hiểu được đóng gói với những nội dung sau:

1. Báo cáo kết quả khoảng 15 trang A4, viết theo kiểu một báo cáo khoa học.
2. Bài trình bày slides khoảng 15 phút giới thiệu toàn bộ những kết quả đã có.

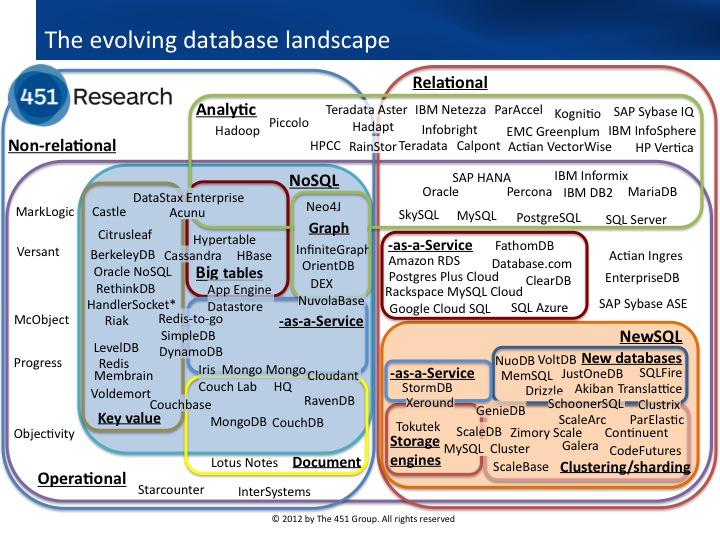
# Giới thiệu Couchbase

Couchbase Server, ban đầu được gọi là Membase, là phần mềm cơ sở dữ liệu định hướng tài liệu đa mô hình NoQuery, được tối ưu hóa cho các ứng dụng tương tác. Các ứng dụng này có thể phục vụ nhiều người dùng đồng thời bằng cách tạo, lưu trữ, truy xuất, tổng hợp, thao tác và trình diễn dữ liệu. Couchbase Server được thiết kế để cung cấp quyền truy cập khóa-giá trị khóa hoặc tài liệu JSON dễ dàng với *độ trễ thấp* và *thông lượng duy trì cao*. Nó được thiết kế để được *phân cụm* từ quy mô nhỏ gốm 1 máy đến các mô hình triển khai quy mô rất lớn trải rộng trên nhiều máy. Couchbase có hỗ trợ phiên bản trên Mobile với đặc điểm có tiêu tốn tài nguyên thấp. Couchbase Mobile là phiên bản để thay thế SQLite đang được sử dụng trên nền tảng di động hiện nay phổ biến bởi các ứng dụng trên Apple Store và Goodle Play store.

Couchbase được viết trên ngôn ngữ C/C++, Go và Erlang (sử dụng cho mô hình phân cụm).

Couchbase nằm ở giữa vùng Key Value và Document Store trong hệ sinh thái DBMS. Các sản phẩm tương tự Couchbase gồm có:

* Document Store: CouchDB, IBM Domino, MongoDB…
* Key Value Store: MemcacheDB, Oracle NoSQL, Dynamo…



# Mục đích và đối tượng chính của DBMS,

## Mục đích

* Giải quyết các bài toán tồn tại của mô hình hệ cơ sở dữ liệu liên hệ (RDBMS)
  + Truy vấn chậm khi cơ sở dữ liệu rất lớn
  + Khó mở rộng và phức tạp triển khai
  + Cập nhật dữ liệu yêu cấu ACID, dẫn tới tốn và lãng phí tài nguyên
  + Không mở rộng trên nhiều node được
  + Cần tạo bảng trước khi import dữ liệu dẫn tới không linh hoạt trong ứng dụng
  + Server chính gặp lỗi dẫn tới toàn hệ thống bị gián đoạn
  + Các hoạt động JOIN dữ liệu nhiều, dẫn tới tiêu tốn tài nguyên lớn

Cơ sở dữ liệu thế hệ tiếp theo chủ yếu giải quyết một số điểm: không liên quan, phân tán, nguồn mở và có thể mở rộng theo chiều ngang. Mục đích ban đầu là các cơ sở dữ liệu quy mô web hiện đại. Phong trào bắt đầu từ đầu năm 2009 và đang phát triển nhanh chóng. Thông thường nhiều đặc điểm áp dụng như: không có lược đồ, hỗ trợ sao chép dễ dàng, API đơn giản, cuối cùng là nhất quán / BASE (không phải ACID), số lượng dữ liệu khổng lồ và hơn thế nữa.

## Đối tượng chính

* Áp dụng cho các ứng dụng mới phát sinh hiện nay với cơ sở dữ liệu lớn, có các đòi hỏi như:
  + Lượng dữ liệu khổng lồ (các dịch vụ có hàng tỷ người dùng như Facebook, Youtube, các dịch vụ mạng xã hội …)
  + Truy vấn cần có kết quả nhanh
  + Truy vấn nhiều hơn cập nhật vào CSDL (ví dụ các dịch vụ tìm kiếm như Google)
  + Chèn và cập nhật không đồng bộ, lược đồ ít/đơn giản
  + Thuộc tính ACID không cần thiết (các truy vấn của các dịch vụ mạng xã hội, tìm kiếm…)
  + Ứng dụng cho di động
  + Ứng dụng cho Internet of things
  + Phát hiện gian lận
  + Digital Communication
  + Quản lý thông tin khách hàng
  + Profile management
* Các công ty phù hợp đã dùng dịch vụ của Couchbase: Google, Facebook, ebay, viber, verizon, General Electric, Cisco, Comcast, …

# Mô hình dữ liệu

Dữ liệu của Couchbase dựa trên nền JSON. Có 3 lợi điểm: đơn giản, nhẹ,đọc hiểu dễ.

Dữ liệu JSON có thể lồng nhau theo cấu trúc, cho phép không cần bảng nối hay tham chiếu như RDBMS.

Couchbase có hỗ trợ ACID (atomicity, consistency, isolation, durability) với thao tác trên một tài liệu cụ thể.

## **Dữ liệu**

Couchbase Server lưu trữ dữ liệu dưới dạng các items. Mỗi items bao gồm một khóa (key), theo đó mục được tham chiếu; và một giá trị liên quan, giá trị sẽ ở dạng nhị phân (binary) hoặc JSON.

**Khóa**

Mỗi giá trị (nhị phân hoặc JSON) được xác định bởi một khóa duy nhất (được dùng như id) và không đổi được sau khi lưu, được định nghĩa bởi người dùng hoặc ứng dụng khi item được lưu lại.

Mỗi khóa đảm bảo 3 điều kiện:

1) chuỗi UTF-8 không có khoảng trắng. Các ký tự đặc biệt, chẳng hạn như (,%, /, "và \_, được chấp nhận.

2) Không quá 250 byte

3) Phải là khóa duy nhất trong Bucket của nó.

**Giá trị**

Kích thước tối đa của một giá trị là 20 MiB. Một giá trị có thể là: nhị phân hoặc JSON

Nhị phân: Bất kỳ hình thức nhị phân đều được chấp nhận. Lưu ý rằng giá trị nhị phân không thể được phân tích (parsed), lập chỉ mục hoặc truy vấn: nó chỉ có thể được truy xuất bằng khóa.

JSON: Một giá trị JSON, được gọi là tài liệu (document), có thể được phân tích, lập chỉ mục và truy vấn. Mỗi tài liệu bao gồm một hoặc nhiều thuộc tính, mỗi thuộc tính có giá trị riêng. Một giá trị thuộc tính có thể là loại cơ bản, chẳng hạn như một số, chuỗi hoặc Boolean; hoặc loại phức tạp, chẳng hạn như một tài liệu nhúng hoặc một mảng.

**Cấu trúc tài liệu**

Một tài liệu JSON mẫu có nội dung như sau:

a1 và a2 là các *thuộc tính* tương ứng có một số đơn và một *chuỗi* là *giá trị* của chúng.

a3 là một thuộc tính có giá trị là một *tài liệu nhúng* bao gồm một *thuộc tính* duy nhất, b1, có giá trị riêng là một mảng gồm ba số.

a4 là một thuộc tính có giá trị là một mảng gồm hai tài liệu, mỗi tài liệu chứa một thuộc tính (c1 hoặc c2) có giá trị riêng tương ứng là một chuỗi và một số.

{

"a1" : number,

"a2" : "string",

"a3" : {

"b1" : [ number, number, number ]

},

"a4" : [

{ "c1" : "string", "c2" : number },

{ "c1" : "string", "c2" : number }

]

}

## Lưu dữ liệu trong Bucket, Bộ nhớ

Dữ liệu trong Couchbase được lưu trữ trong các Bucket; được giữ trong bộ nhớ và đĩa. Bucket được tạo sử dụng Couchbase web console, API REST hoặc CLI. Mỗi Bucket được gán một tên khi khởi tạo và được tham chiếu qua truy vấn bởi ứng dụng hoặc người dùng.

Các loại Bucket được sử dụng thường xuyên nhất là Couchbase và Ephemeral. Couchbase bucket tồn tại cả trong bộ nhớ và trên đĩa. Ephemeral Bucket chỉ tồn tại trong bộ nhớ. Các Bucket có thể được cấu hình để duy trì dữ liệu ở định *dạng nén*, để *tối đa hóa* hiệu quả tài nguyên. Các tài liệu trong một Bucket có thể được chỉ định *Thời gian lưu*, để đảm bảo rằng chúng hết hạn sau một khoảng thời gian xác định.

Couchbase Server quản lý bộ nhớ để đảm bảo hiệu suất và khả năng mở rộng: hạn ngạch bộ nhớ được thiết lập và dữ liệu không được sử dụng trong thời gian gần có thể được đẩy ra, để nhường chỗ cho dữ liệu được yêu cầu thường xuyên hơn. Các trình đọc và ghi *đa luồng* cung cấp các hoạt động đọc và ghi đồng thời cho dữ liệu trên đĩa, đảm bảo thông lượng cao.

## Ngôn ngữ truy vấn dữ liệu

Couchbase sử dụng N1QL, ngôn ngữ có cấu trúc tương tự SQL.

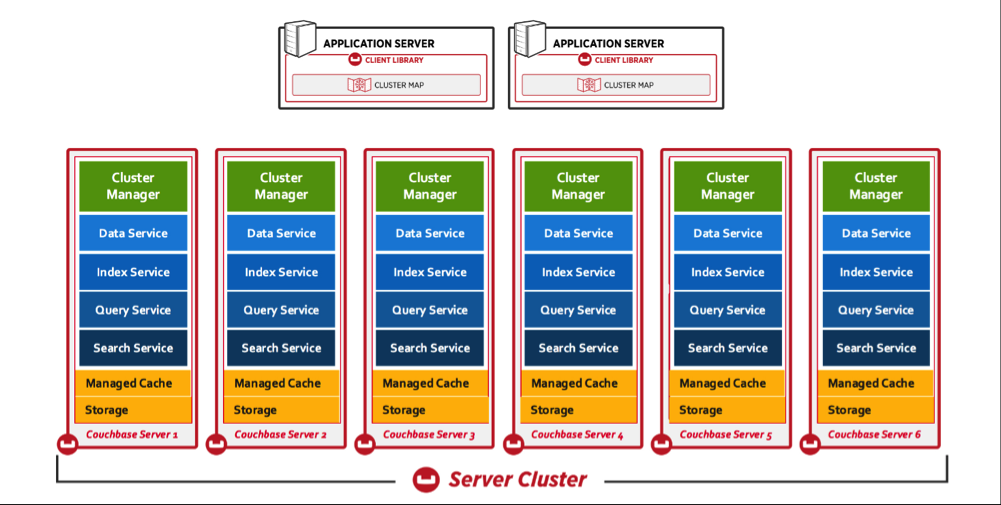
N1QL hỗ trợ nhiều loại dữ liệu: Boolean, số, string, mảng, objects, NULL, và MISSING. Tuy nhiên không hỗ trợ loại dữ liệu Date, nhưng cho phép có mộ bộ tính năng làm việc với ngày tháng.

|  |  |
| --- | --- |
| **Kiểu dữ liệu trong N1QL** | **Đặc điểm** |
| Boolean | Loại này có thể có giá trị TRUE và FALSE. Các giá trị không phân biệt chữ hoa chữ thường. |
| Số | Các số có thể là một số thập phân có chữ ký có thể chứa một phần phân số. |
| Chuỗi | Chuỗi bao gồm tất cả các ký tự Unicode và ký tự thoát dấu gạch chéo ngược, được phân cách bằng dấu ngoặc kép đơn ('') hoặc dấu ngoặc kép (""). |
| Mảng | Mảng là một danh sách được sắp xếp từ 0 hoặc nhiều giá trị. Các giá trị có thể là bất kỳ loại nào. Mảng được đặt trong dấu ngoặc vuông ([]). Dấu phẩy riêng biệt từng giá trị.  Ví dụ: ["một", "hai", "ba"] và [1, 2, 3] và ["một", 2, "ba"].  Các đối tượng (Objects)  Các đối tượng bao gồm các cặp tên-giá trị. Tên phải là một chuỗi và giá trị có thể là bất kỳ loại dữ liệu JSON được hỗ trợ nào.  Các đối tượng được đặt trong dấu ngoặc nhọn ({}). Dấu phẩy riêng biệt từng cặp. Ký tự dấu hai chấm (:) tách tên khỏi giá trị của nó trong mỗi cặp.  Dưới đây là một số ví dụ về đối tượng: {"age": 17}, {"fname": "Jane", "lname": "Doe"}  Tất cả các tên phải là các chuỗi và phải khác biệt với nhau trong đối tượng đó. |
| NULL | Nulls đại diện cho các giá trị trống bằng cách sử dụng từ khóa NULL. Ví dụ: nhà phát triển ban đầu có thể đặt giá trị trường thành null theo mặc định cho đến khi người dùng nhập giá trị.  Các giá trị Null cũng được tạo bởi các hoạt động nhất định, ví dụ, khi chia cho 0 hoặc chuyển các đối số của loại sai.  Giá trị NULL trong N1QL không phân biệt chữ hoa chữ thường. Ví dụ: null, NULL, Null và nUll đều tương đương nhau. |
| MISSING | MISSING biểu thị một cặp tên-giá trị bị thiếu trong tài liệu JSON. Nếu trường tham chiếu không tồn tại, giá trị kết quả trống được truy vấn trả về. MISSING được thêm vào như một biểu thức nghĩa đen, mặc dù nó không được trả lại trong kết quả cuối cùng. Nó được bỏ qua từ các đối tượng và được chuyển đổi thành null trong mảng kết quả  Vì N1QL không bị ràng buộc bởi một lược đồ cố định, một số đối tượng và tài liệu có thể chứa các trường mà các trường khác không chứa; vì vậy một trường có thể có mặt trong một tài liệu và MISSING trong một tài liệu khác. (MISSING không có trong SQL, vì mọi bản ghi trong bảng đều tuân theo một lược đồ giống nhau.) |
| Đối chiếu (Collation) | Đối chiếu xác định thứ tự sắp xếp cho các loại dữ liệu. Dưới đây là thứ tự đối chiếu được sử dụng cho các loại dữ liệu N1QL:  MISSING, Null, Boolean, Number, String, Array, Objects, binary (non JSON) |
| Ngày | N1QL không hỗ trợ kiểu dữ liệu Ngày; tuy nhiên N1QL không cung cấp đầy đủ các chức năng để có thể sử dụng để thao tác ngày. Hỗ trợ trong hàm *Date Functions* |
| Nhị phân | Khi Couchbase Server được sử dụng làm kho lưu trữ khóa-giá trị và giá trị là giá trị không phải là JSON, giá trị được hiển thị dưới dạng nhị phân. N1QL không phân tích dữ liệu nhị phân. Giá trị trả về chỉ biểu thị rằng giá trị là giá trị nhị phân. Ví dụ: "\ u003cbinary (size\_in\_bytes b) \ u003e", trong đó size\_in\_bytes biểu thị kích thước của dữ liệu nhị phân theo byte. Có thể sử dụng hàm TYPE () để truy xuất loại giá trị dưới dạng nhị phân. Các hàm trả về siêu dữ liệu (metadata) như META () vẫn hoạt động như mong đợi. Tuy nhiên, hoạt động trên dữ liệu nhị phân bị hạn chế |

# Phương pháp kỹ thuật xử lý giao tác đồng thời

## Kiến **trúc xử lý**

### Mô hình cluster



Couchbase triển khai theo mô hình cụm server, mỗi server gồm 3 thành phần Cluster Manager, Service và Storage

* Cluster manager: trình quản lý cụm
* Service: gồm 4 dịch vụ: dữ liệu, chỉ mục, truy vấn và tìm kiếm
* Cache và Storage: lưu trữ

Dịch vụ *dữ liệu (Data Service)*: cung cấp quyền truy cập vào dữ liệu trong bộ nhớ và trên đĩa.

Dịch vụ *truy vấn (Query Service)*: hỗ trợ truy vấn dữ liệu bằng ngôn ngữ truy vấn **N1QL**.

Dịch vụ *chỉ mục (Index Service):* Couchbase hỗ trợ tạo các *chỉ mục (index)* chính và phụ trên các mục được lưu trữ trong Couchbase Server.

Chương trình *quản lý cụm*: Trình quản lý cụm Couchbase chạy trên tất cả các node của cụm, duy trì các quy trình trên mỗi node cần thiết và điều phối các hoạt động trên toàn cụm.

*Tính khả dụng của cụm* : Trên các node của mỗi cụm, dữ liệu Couchbase được phân phối và nhân rộng đồng đều: các node có thể được loại bỏ và xử lý lỗi mà không mất dữ liệu. Dữ liệu có thể được chọn để nhân rộng trên các cụm cư trú trong các trung tâm dữ liệu khác nhau (Cross Datacenter), để đảm bảo tính sẵn sàng cao (HA).

**Dịch vụ**

Các dịch vụ được cấu hình và triển khai bởi Quản trị viên trên một hoặc nhiều nút. Mỗi dịch vụ hỗ trợ một hình thức truy cập dữ liệu cụ thể.

*Dịch vụ* có thể được triển khai để hỗ trợ các hình thức truy cập dữ liệu khác nhau: ví dụ: *Dịch vụ dữ liệu* cho phép các mục được truy xuất bằng khóa; trong khi *Dịch vụ truy vấn* cho phép truy vấn dữ liệu thông qua phương tiện bằng ngôn ngữ truy vấn N1QL(tương tự SQL). Các dịch vụ riêng lẻ có thể được cấu hình để chạy trên nhiều nút cụm (cluster), cho phép khối lượng công việc ưu tiên cao được phân phối và tán nhỏ một cách thích hợp. Lập chỉ mục các dịch vụ hỗ trợ, bằng cách cho phép truy cập dữ liệu hiệu suất cao.

**Chỉ mục**

Các chỉ mục được sử dụng bởi một số dịch vụ nhất định, chẳng hạn như Truy vấn, Phân tích và Tìm kiếm. Chỉ mục là tập con được xác định trước của dữ liệu (bucket-data) phục vụ cho tìm kiếm. Chỉ mục khi cho phép các cải tiến đáng kể cho hiệu suất của hoạt động tìm kiếm.

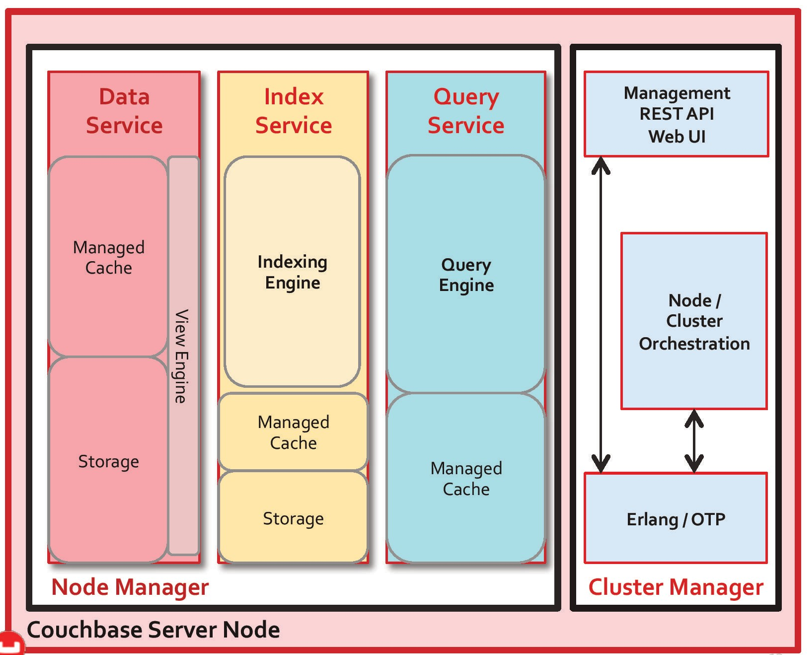
Cả hai dịch vụ **Truy vấn** và **Phân tích** đều dựa vào các chỉ mục được cung cấp bởi dịch vụ Chỉ mục (Index Service).

*Chỉ mục chính* (primary index): Được cung cấp bởi *Dịch vụ Chỉ mục*, dựa trên khóa duy nhất của mỗi mục trong một nhóm được chỉ định. Mỗi chỉ mục chính được duy trì không đồng bộ. Một chỉ mục chính được thiết kế để dùng cho các truy vấn đơn giản, không có bộ lọc hoặc vị từ.

*Chỉ mục thứ cấp*: Được cung cấp bởi Dịch vụ Chỉ mục. Giá trị được liên kết với thuộc tính có thể thuộc bất kỳ loại nào: vô hướng, đối tượng hoặc mảng.

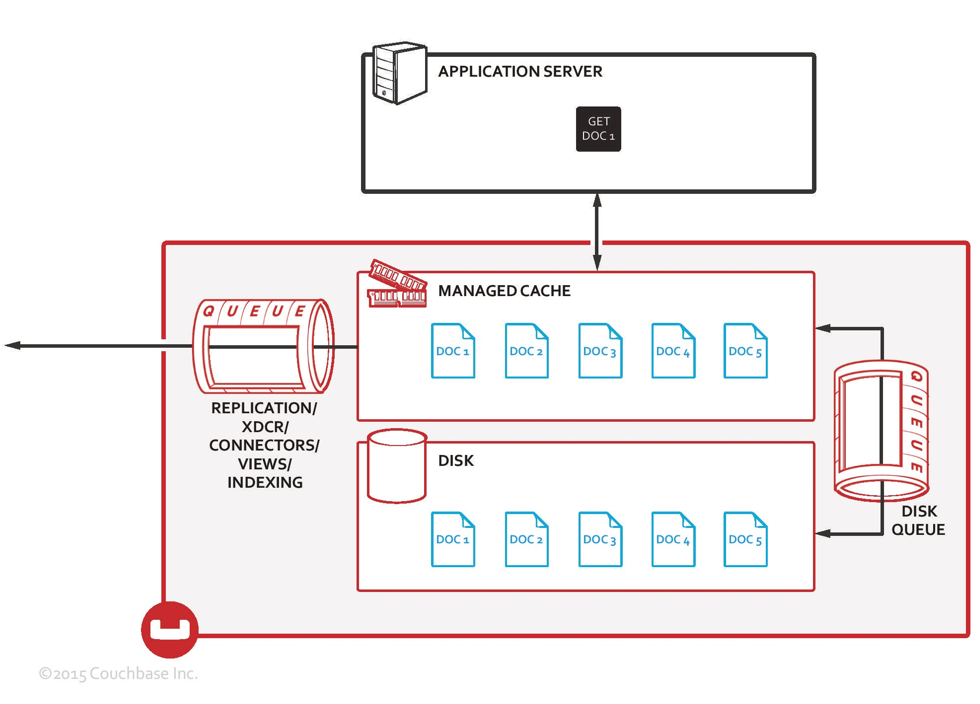
*Chỉ số phụ* GSI: Đây là loại chỉ mục được sử dụng thường xuyên nhất trong các Máy chủ Couchbase, cho các truy vấn được thực hiện với ngôn ngữ truy vấn N1Ql.

### Mô hình của một node:



### Truy vấn dữ liệu

Couchbase truy vấn dữ liệu từ bộ nhớ cache (RAM). Cho phép xử lý song song, đọc dữ liệu từ RAM, đồng bộ dữ liệu từ đĩa lên RAM.



Luồng truy vấn: gồm 8 bước

1/ Submit query

2/ Phân tích

3/ Quét, lọc index

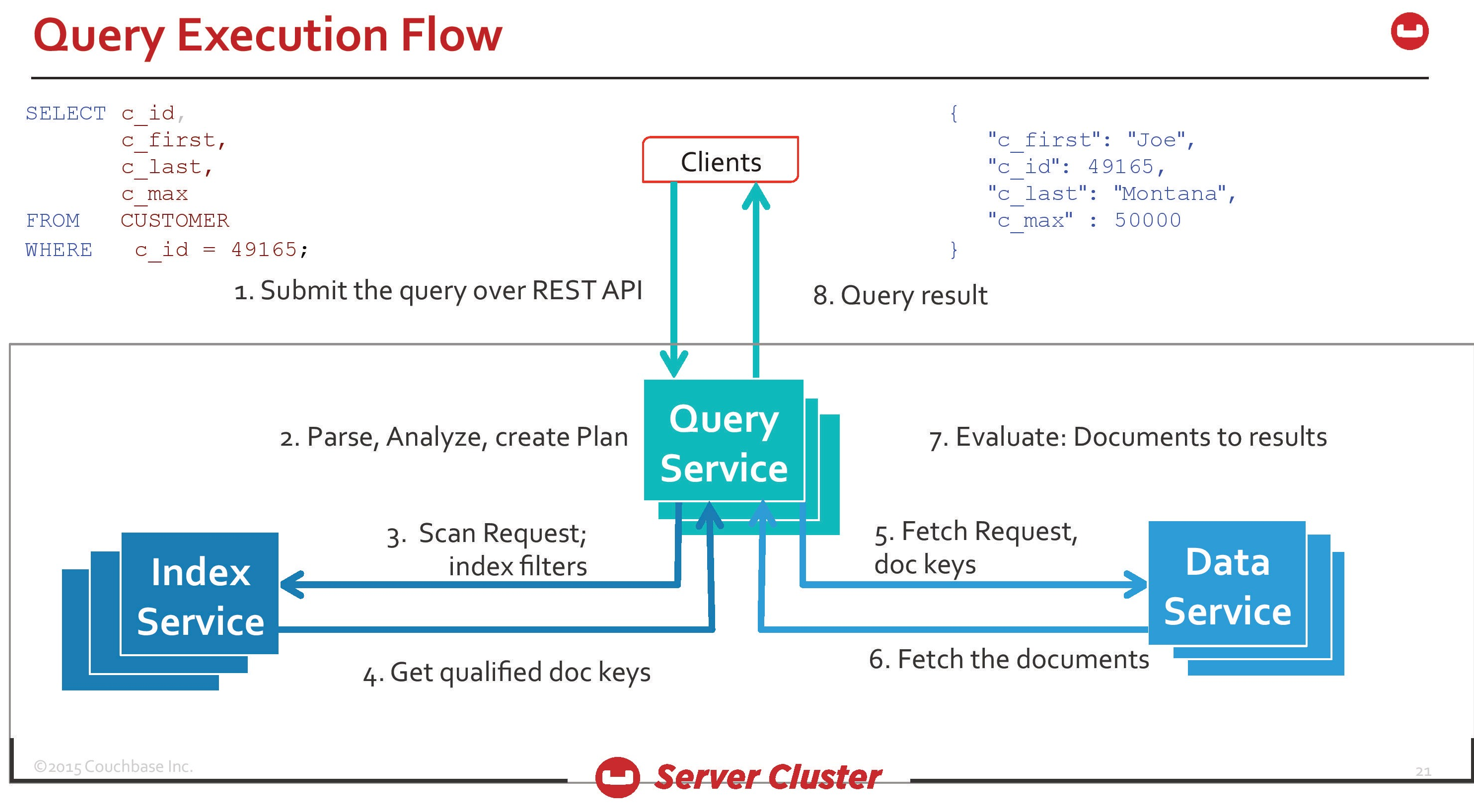
4/ Tìm khóa

5/ Yêu cầu theo dockey

6/ Lấy dữ liệu

7/ Kiểm tra

8/ Hiển thị dữ liệu



# Hiệu năng, so sánh và đánh giá

## Hiệu suất

### Import dữ liệu

* Sử dụng công cụ cbimport được couchbase khuyến nghị với cú pháp chuẩn, đã tăng số luồng tối đa tương ứng số nhân của chip xử

Kết quả, import bảng dữ liệu 60 triệu dòng: 1 tiếng/3 triệu dòng, 6 tiếng trên 20 triệu dòng

|  |  |  |
| --- | --- | --- |
|  | **Couchbase** | **Sqlite** |
| Loại DB | NoSQL | RDBMS |
| Số dòng dữ liệu import | 1.401.452 | 9.073.533 |
| Tài nguyên cần thiết | ~ 200 MB ram / 1 triệu bản ghi | Không quản lý dữ liệu trên ram, query vào ổ cứng |
| Thời gian import | 10 phút | 2 phút |
| Cú pháp | cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket namebasics -t 4 --dataset file:\\d:\C\Imdb\tsv\namebasics.csv --field-separator \t --username Administrator --password 123456 --generate-key key::nconst::#MONO\_INCR# | sqlite> .import namebasics.csv namebasics  sqlite> select count(nconst) from namebasics;  9073552 |
| Nhận xét | Couchbase cung cấp công cụ cbimport trên phân hệ linux (Ubuntu, Centos, Debian) và Windows. Sử dụng option -t (thread 4 luồng) để tăng tốc tối đa quá trình import dữ liệu.  Ram không ảnh hưởng nhiều, CPU cao chiếm 50% CPU utilization, tổng CPU đạt 85% | Không cấu hình, quá trình import chỉ ảnh hưởng ít tới ram, CPU usage khoảng 75-85% |
|  |  | 63.523.489 |
| Câu lệnh |  | sqlite> .import titleprincipal(mul2).csv titleprincipals  titleprincipal(mul2).csv:63523489: expected 6 columns but found 1 - filling the rest with NULL  sqlite> |
| Thời gian |  | 15 phút |
|  |  |  |

### Truy vấn dữ liệu

##### Count

|  |  |  |
| --- | --- | --- |
|  | **Couchbase** | **Sqlite** |
| Số dòng dữ liệu đưa vào đếm | 1.500.000 | 1.500.000 |
| Index | Đã index trước khi import | Không index |
| Tài nguyên cần thiết | Sử dụng ram được cấp phát cho dịch vụ | Không tiêu thụ nhiều RAM |
|  |  |  |
| Thời gian đếm | 8 phút | 1 giây |
| Cú pháp | select count(nconst) from (select nconst from `titleprincipal-60mil`limit 1000000) as nconst;  success | elapsed: 8.00m | execution: 8.00m | count: 1 | size: 37 | sqlite> select count(nconst) from (select nconst from titleprincipals limit 1000000); |
| Kết quả | [  {  "$1": 1000000  }  ] | 1000000 |
|  |  |  |
| Số dòng dữ liệu đưa vào đếm |  | 37 triệu dòng |
| Thời gian đếm |  | 25 giây |
| Cú pháp |  | sqlite> select count(titleID) from titleakas; |
| Kết quả |  | 37.269.541 |
|  |  |  |
| Số dòng dữ liệu đưa vào đếm |  | 62 triệu dòng |
| Thời gian đếm |  | 33 giây |
| Cú pháp |  | sqlite> select count(tconst) from titleprincipals; |
| Kết quả |  | 63.523.489 |

##### Join 2 bảng

|  |  |  |
| --- | --- | --- |
|  | Couchbase | Sqlite |
| Số dòng dữ liệu đưa vào đếm | 1.4 triệu bản ghi từ bảng titleprincipals  382k bản ghi từ bảng titleakas  Tổng 1.8 triệu bản ghi | 60 triệu bản ghi từ bảng titleprincipals (được x2)  37 triệu bản ghi từ bảng titleakas (được x10 so với dữ liệu gốc)  Tổng 97 triệu bản ghi |
| Index | Đã index trước khi import | Không index |
| Tài nguyên cần thiết | Sử dụng ram được cấp phát cho dịch vụ | Không tiêu thụ nhiều RAM |
| Thời gian đếm | NA | 12 phút |
| Câu lệnh | select `titleprincipal-60mil`.nconst nconst, `titleprincipal-60mil`.tconst tconst, `titleakas-37mil`.titleId titleId, `titleakas-37mil`.title title from `titleprincipal-60mil` left join `titleakas-37mil` on `titleprincipal-60mil`.tconst = `titleakas-37mil`.titleId limit 10; | sqlite> select titleprincipals.nconst nconst, titleprincipals.tconst tconst, titleakas.titleId titleId, titleakas.title title from titleprincipals left join titleakas on titleprincipals.tconst = titleakas.titleId limit 10; |
| Kết quả | [  {  "code": 4330,  "msg": "No index available for ANSI join term titleakas-37mil",  "query\_from\_user": "select `titleprincipal-60mil`.nconst nconst, `titleprincipal-60mil`.tconst tconst, `titleakas-37mil`.titleId titleId, `titleakas-37mil`.title title from `titleprincipal-60mil` left join `titleakas-37mil` on `titleprincipal-60mil`.tconst = `titleakas-37mil`.titleId limit 10;"  }  ] | nconst,tconst,,  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  nm1588970,tt0000001,tt0000001,Carmencita  sqlite> |

select count (KI) from nhanvien, bảng kích thước 800 nghìn dòng

success | elapsed: 5.00m | execution: 5.00m | count: 1 | size: 36

select count (distinct KI) from nhanvien

tạo index

CREATE PRIMARY INDEX `tconst` ON `titleprincipals-60mil`

### Cập nhật dữ liệu

cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket titleprincipals-60mil -t4 --dataset file:\\d:\C\Imdb\tsv\titleprincipal(mul2).csv --username Administrator --password 123456 --generate-key key::tconst::#MONO\_INCR#

### Chèn

### Dữ liệu các bảng

more /e "titleakas(mul5).csv"

titleId ordering title region language types attributes isOriginalTitle

tt0000001 1 Carmencita - spanyol t├ínc HU \N imdbDisplay \N 0

tt0000001 2 ╬Ü╬▒╧ü╬╝╬╡╬╜╧â╬»╧ä╬▒ GR \N \N \N 0

tt0000001 3 ╨Ü╨░╤Ç╨╝╨╡╨╜╤ü╨╕╤é╨░ RU \N \N \N 0

tt0000001 4 Carmencita US \N \N \N 0

tt0000001 5 Carmencita \N \N original \N 1

$ more /e "titleprincipal(mul2).csv"

tconst ordering nconst category job characters

tt0000001 1 nm1588970 self \N ["Herself"]

tt0000001 2 nm0005690 director \N \N

tt0000001 3 nm0374658 cinematographer director of photography \N

tt0000002 1 nm0721526 director \N \N

tt0000002 2 nm1335271 composer \N \N

> more/e namebasics.tsv

nconst primaryName birthYear deathYear primaryProfession knownForTitles

nm0000001 Fred Astaire 1899 1987 soundtrack,actor,miscellaneous tt0050419,tt0053137,tt0043044,tt0072308

nm0000002 Lauren Bacall 1924 2014 actress,soundtrack tt0071877,tt0117057,tt0037382,tt0038355

nm0000003 Brigitte Bardot 1934 \N actress,soundtrack,producer tt0049189,tt0057345,tt0059956,tt0054452

nm0000004 John Belushi 1949 1982 actor,writer,soundtrack tt0077975,tt0072562,tt0080455,tt0078723

nm0000005 Ingmar Bergman 1918 2007 writer,director,actor tt0050976,tt0050986,tt0060827,tt0083922

nm0000006 Ingrid Bergman 1915 1982 actress,soundtrack,producer tt0034583,tt0038109,tt0038787,tt0071877

nm0000007 Humphrey Bogart 1899 1957 actor,soundtrack,producer tt0037382,tt0043265,tt0034583,tt0033870

nm0000008 Marlon Brando 1924 2004 actor,soundtrack,director tt0068646,tt0078788,tt0047296,tt0070849

nm0000009 Richard Burton 1925 1984 actor,producer,soundtrack tt0059749,tt0057877,tt0061184,tt0087803

## Đánh giá

Couchbase gặp khó khăn khi triển khai mô hình 1 node với các thao tác cơ bản

* Import dữ liệu
* Query dữ liệu

> cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket titleakas-37mil -t 4 --dataset file:\\d:\C\Imdb\tsv\titleakas(mul10).csv --field-separator \t --username Administrator --password 123456 --generate-key key::titleId::#MONO\_INCR#

2019-01-18T21:01:55.015+07:00 ERRO: line 392427, column 27: bare " in non-quoted-field -- plan.(\*data).execute() at data.go:89

CSV import failed: 383159 documents were imported, 0 documents failed to be imported

CSV import failed: line 392427, column 27: bare " in non-quoted-field

tungd@DESKTOP-VE2C3MU c:\Program Files\Couchbase\Server\bin

> cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket titleprincipal-60mil -t 1 --dataset file:\\d:\C\Imdb\tsv\titleprincipal(mul2).csv --field-separator \t --username Administrator --password 123456 --generate-key key::tconst::#MONO\_INCR#

2019-01-19T01:49:49.692+07:00 ERRO: Timeout occurred (180 sec) on operation -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

2019-01-19T01:49:50.103+07:00 ERRO: Timeout occurred (180 sec) on operation -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

2019-01-19T01:49:50.446+07:00 ERRO: Timeout occurred (180 sec) on operation -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

2019-01-19T02:34:09.814+07:00 WARN: Invalid number of fields on line 63523489: Expected 6, got 1 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-19T02:34:10.050+07:00 ERRO: Not all data was sent to Couchbase due to connectivity issues. Check to make sure there were no server side failures. See the logs for for more details on what wasn't sent. -- plan.(\*data).execute() at data.go:89

CSV import failed: **63523487** documents were imported, 1 documents failed to be imported

CSV import failed: Not all data was sent to Couchbase due to connectivity issues. Check to make sure there were no server side failures. See the logs for for more details on what wasn't sent.

# Tài liệu cài đặt và cấu hình

## Couchbase

### Cài đặt:

Cài đặ trên centos 7

Bước 1: Kiểm tra cài đặt gói pkgconfig

sudo yum install -y pkgconfig

Bước 2: Kiểm tra cài đặt open ssl

rpm -q -a | grep "openssl"

### Địa chỉ truy cập server

### Tạo cluster, cấu hình tài nguyên cho cluster

Truy xuất Couchbase workbench từ địa chỉ <http://127.0.0.1:8091>

Nhập user/pass quả trị đã cấu hình trong quá trình cài đặt

### Tạo bucket, cấu hình tài nguyên cho bucket

### Tạo index

CREATE INDEX index\_name ON named\_keyspace\_ref ( expression1 [ , expression2 ] \* )

WHERE filter\_expressions

[ USING GSI | VIEW ]

[ WITH { "nodes": [ "node\_name" ], "defer\_build":true|false , "num\_replica": num\_replica\_num } ];

CREATE INDEX over5 ON `beer-sample`(abv) WHERE abv > 5 USING GSI WITH {"nodes": ["192.0.2.1:8091"]};

CREATE INDEX t1 ON `titleprincipals`(seasonNumber) WHERE seasonNumber > 0 USING GSI WITH {"nodes": ["127.0.0.1:8091"]};

Syntax

CREATE INDEX index\_name ON named\_keyspace\_ref ( expression1 [ , expression2 ] \* )

WHERE filter\_expressions

[ USING GSI | VIEW ]

[ WITH { "nodes": [ "node\_name" ], "defer\_build":true|false , "num\_replica": num\_replica\_num } ];

CREATE PRIMARY INDEX `beer\_primary` ON `beer-sample` WITH { "defer\_build":true }

### Import dữ liệu

**Câu lệnh**

cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket nhanvien --dataset file:\\d:\C\Imdb\nhan\_vien.csv --username Administrator --password 123456 --generate-key key::%Ma\_NV%::#MONO\_INCR#

**báo import thành công**

C:\Program Files\Couchbase\Server\bin>cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket nhanvien --dataset file:\\f:\COUCH\ki\_table\_1.csv --username Administrator --password 123456 --generate-key key::Ma\_NV::#MONO\_INCR#

CSV `file:\\f:\COUCH\ki\_table\_1.csv` imported to `http://127.0.0.1:8091` successfully, 1000000 documents loaded

C:\Program Files\Couchbase\Server\bin>cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket nhanvien --dataset file:\\f:\COUCH\ki\_table\_2.csv --username Administrator --password 123456 --generate-key key::Ma\_NV::#MONO\_INCR#

CSV `file:\\f:\COUCH\ki\_table\_2.csv` imported to `http://127.0.0.1:8091` successfully, 1000000 documents loaded

**Các lỗi thường xảy ra khi import**

* **Chỉ định đường dẫn**: file://

CSV import failed: 0 documents were imported, 0 documents failed to be imported

CSV import failed: Invalid scheme error for `d:\C\Imdb\tsv\titleepisode.tsv`, expected file://

cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket nhanvien -t 4 --dataset file://d:\C\Imdb\ki\_table\_99.csv --username Administrator --password 123456 --generate-key key::Ma\_NV::#MONO\_INCR#

2019-01-17T21:20:59.660+07:00 ERRO: parse file://d:\C\Imdb\ki\_table\_99.csv: invalid character "\\" in host name -- plan.(\*data).execute() at data.go:89

CSV import failed: 0 documents were imported, 0 documents failed to be imported

CSV import failed: parse file://d:\C\Imdb\ki\_table\_99.csv: invalid character "\\" in host name

* **Không đủ bộ nhớ**

[9:43:29 PM] - Metadata overhead warning. Over 59% of RAM allocated to bucket "nhanvien" on node "127.0.0.1" is taken up by keys and metadata.

2019-01-17T21:49:02.698+07:00 ERRO: server is out of memory -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

2019-01-17T21:49:05.891+07:00 ERRO: server is out of memory -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

[9:43:29 PM] - Metadata overhead warning. Over 59% of RAM allocated to bucket "nhanvien" on node "127.0.0.1" is taken up by keys and metadata.   
[9:49:05 PM] - Hard out-of-memory error: Bucket "nhanvien" on node 127.0.0.1 is full. No memory currently allocated to this bucket can be easily released.

* **Time out**

> cbimport.exe csv --cluster couchbase://127.0.0.1 --bucket nhanvien -t 4 --limit-rows 1000002 --dataset file:\\d:\C\Imdb\ki\_table\_99.csv --username Administrator --password 123456 --generate-key key::Ma\_NV::#MONO\_INCR#

2019-01-17T21:39:25.091+07:00 ERRO: Timeout occurred (180 sec) on operation -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

2019-01-17T21:39:49.535+07:00 ERRO: Timeout occurred (180 sec) on operation -- couchbase.(\*Callbacks).handleError() at callbacks.go:117

2019-01-17T21:43:21.098+07:00 ERRO: Not all data was sent to Couchbase due to connectivity issues. Check to make sure there were no server side failures. See the logs for for more details on what wasn't sent. -- plan.(\*data).execute() at data.go:89

CSV import failed: 993335 documents were imported, 0 documents failed to be imported

CSV import failed: Not all data was sent to Couchbase due to connectivity issues. Check to make sure there were no server side failures. See the logs for for more details on what wasn't sent.

* **Kết nối quản trị gặp khó khăn do thiếu bộ nhớ**

Difficulties communicating with the cluster. Displaying cached information.

Tạo index

Cú pháp chuẩn

CREATE INDEX index\_name ON named\_keyspace\_ref ( expression1 [ , expression2 ] \* )

WHERE filter\_expressions

[ USING GSI | VIEW ]

[ WITH { "nodes": [ "node\_name" ], "defer\_build":true|false , "num\_replica": num\_replica\_num } ];

Ví dụ

CREATE INDEX over5 ON `beer-sample`(abv) WHERE abv > 5 USING GSI WITH {"nodes": ["192.0.2.1:8091"]};

CREATE INDEX t1 ON `titleprincipals`(seasonNumber) WHERE seasonNumber > 0 USING GSI WITH {"nodes": ["127.0.0.1:8091"]};

* Dữ liệu không đúng định dạng

5+07:00 WARN: Invalid number of fields on line 2474: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:39.862+07:00 WARN: Invalid number of fields on line 2475: Expected 1, got 3 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:39.986+07:00 WARN: Invalid number of fields on line 2483: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:40.148+07:00 WARN: Invalid number of fields on line 2511: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:40.422+07:00 WARN: Invalid number of fields on line 2524: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:40.642+07:00 WARN: Invalid number of fields on line 2551: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:40.813+07:00 WARN: Invalid number of fields on line 2552: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:41.042+07:00 WARN: Invalid number of fields on line 2559: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:41.259+07:00 WARN: Invalid number of fields on line 2586: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:41.633+07:00 WARN: Invalid number of fields on line 2609: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

2019-01-18T14:26:41.870+07:00 WARN: Invalid number of fields on line 2655: Expected 1, got 2 -- csvdata.(\*csvSource).Data() at source.go:235

### Tắt/bật dịch vụ server

* Chạy cmd với Administrator Previlege: net stop couchbaseserver
* Kết quả: The CouchbaseServer service is stopping..

**Cấu trúc câu query sử dụng n1ql của Couchbase**

## Sqlite3

## mysql

### Cài đặt:

tham khảo hướng dẫn trên trang chủ mysql cho phiên bản windows, và linux

### Kết nối tới server:

kết nối từ giao diện dòng lệnh Mysql Shell

\connect [root@127.0.0.1](mailto:root@127.0.0.1)

### Tạo db

### Sử dụng DB:

use imdb

### Kiểm tra DB

Sử dụng lệnh ‘show databases’

MySQL 127.0.0.1:33060+ ssl imdb SQL > show databases;

+--------------------+

| Database |

+--------------------+

| imdb |

| information\_schema |

| mysql |

| performance\_schema |

| sakila |

| sys |

| world |

+--------------------+

7 rows in set (0.0028 sec)

### Tạo bảng

create table titleprincipals (tconst varchar(15), ordering varchar(5), nconst varchar(15), category varchar(50), job varchar(100), characters varchar(255));

### Kiểm tra bảng

Dùng 2 lệnh `show` và `describe` để kiểm tra bảng

show tables;

+-----------------+

| Tables\_in\_imdb |

+-----------------+

| titleprincipals |

+-----------------+

1 row in set (0.0079 sec)

describe titleprincipals;

+------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| tconst | varchar(15) | YES | | NULL | |

| ordering | varchar(5) | YES | | NULL | |

| nconst | varchar(15) | YES | | NULL | |

| category | varchar(50) | YES | | NULL | |

| job | varchar(100) | YES | | NULL | |

| characters | varchar(255) | YES | | NULL | |

+------------+--------------+------+-----+---------+-------+

### Cấu hình cho phép import dữ liệu local

Lệnh: *set local infile on*

SET GLOBAL local\_infile = 'ON';

Kiểm tra kết quả: MySQL 127.0.0.1:33060+ ssl SQL > show variables like '%local%';

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| local\_infile | **ON** |

+---------------+-------+

1 row in set (0.0240 sec)

### Import dữ liệu vào bảng

Lệnh: load data infile "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/titleprincipals.tsv" into table titleprincipals;

### Tắt dịch vụ/bật dịch vụ mySQL Server

C:\> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld"

C:\> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqladmin" -u root shutdown

# Tài liệu tham khảo

<https://en.wikipedia.org/wiki/ACID_(computer_science)>

<http://en.wikipedia.org/wiki/NoSQL>

<http://nosql-database.org/>

<https://docs.couchbase.com>

<https://dev.mysql.com>

<https://www.networkworld.com/article/2999856/big-data-business-intelligence/10-use-cases-where-nosql-will-outperform-sql.html>