

BÀI TẬP LỚN 2

CƠ CHẾ PHÂN TÁN TRÊN HỆ QUẢN TRỊ NOSQL: Couchbase

Môn: Cơ sở dữ liệu phân tán
Lớp: **IS211.M11.2** - Nhóm 10

Phạm Quốc Hùng	19521579	19521579@gm.uit.edu.vn
Chu Hà Thảo Ngân	19521882	19521882@gm.uit.edu.vn
Thái Minh Triết	19522397	19522397@gm.uit.edu.vn
Võ Tuấn Anh	19521226	19521226@gm.uit.edu.vn

THÁNG 12/2021

Mục lục

<i>Yêu cầu 1: Giới thiệu về hệ quản trị cơ sở dữ liệu NoSQL: Couchbase</i>	3
1. Lịch sử ra đời và nguồn gốc	3
2. Tạo và thêm mới dữ liệu	4
2.1. Tải Couchbase	4
2.2. Thực hiện cài đặt Couchbase	6
2.3. Thực hiện cài đặt Cluster (cụm) mới	8
2.4. Thực hiện thêm dữ liệu	11
2.4.1. Nạp và sử dụng dữ liệu có sẵn	11
2.4.2. Thêm dữ liệu mới	14
2.4.2.1. Cách tổ chức dữ liệu của Couchbase	14
2.4.2.2. Thực hiện tạo mới dữ liệu bằng giao diện Couchbase	15
2.4.2.3. Thực hiện tạo mới dữ liệu trên Couchbase bằng Python	20
<i>Yêu cầu 2, 3: Cơ chế truy vấn và thao tác dữ liệu phân tán</i>	23
1. Mô tả dữ liệu mẫu: Travel App	23
2. Truy vấn dữ liệu	25
2.1. Truy vấn dữ liệu trực tiếp bằng Query Workbench của Couchbase dùng N1QL Query	25
2.2. Truy vấn dữ liệu trực tiếp bằng Python không dùng N1QL Query	28
2.3. Truy vấn dữ liệu bằng Python dùng N1QL Query	29
3. Cài đặt trên nhiều máy	32
4. Lấy dữ liệu qua lại giữa hai máy bằng NoSQL	35
4.1. Yêu cầu trước khi thực hiện truy vấn phân tán	35
4.2. Truy vấn dữ liệu phân tán bằng Query Workbench của Couchbase dùng N1QL Query	35
4.3. Truy vấn dữ liệu phân tán bằng Python dùng N1QL Query	36
5. Thao tác thêm, xóa, sửa dữ liệu giữa hai máy bằng NoSQL	38
5.1. Thao tác thêm dữ liệu	38
5.2. Thao tác xóa dữ liệu	39
5.2.1. Thao tác xóa dữ liệu có điều kiện	39

5.2.2. Thao tác xoá dữ liệu sử dụng ID của document	39
5.3. Thao tác sửa dữ liệu	40
5.3.1. Thao tác sửa dữ liệu dùng ID document	40
5.3.2. Thao tác sửa dữ liệu dùng Sub-query	40
Phụ lục:	41
Tham khảo	41
Các đường dẫn liên quan	41

Yêu cầu 1:

Giới thiệu về hệ quản trị cơ sở dữ liệu NoSQL: Couchbase

1. Lịch sử ra đời và nguồn gốc



Hình 1. Logo Couchbase

Couchbase Server, từng được gọi là Membase, là một gói phần mềm cơ sở dữ liệu phân tán hướng tài liệu NoSQL đa mô hình được tối ưu cho các ứng dụng tương tác. Những ứng dụng này có thể phục vụ đồng thời nhiều người dùng trong việc tạo mới, lưu trữ, truy xuất và tích hợp. Couchbase Server được thiết kế để truy cập các document dạng JSON hay key-value với độ trễ thấp và duy trì thông lượng ở mức cao.

Membase được phát triển bởi những người đứng đầu dự án memcached, những người đã sáng lập công ty NorthScale. Mục tiêu của dự án là phát triển một kho lưu trữ dạng key-value với sự đơn giản, tốc độ và khả năng mở rộng của memcached, cùng với đó là khả năng lưu trữ, tính toàn vẹn và khả năng truy vấn của một cơ sở dữ liệu. Source code ban đầu của dự án được đóng góp bởi NorthScale, Zynga and Naver Corporation trên membase.org vào tháng 6 năm 2010.

Ngày 8 tháng 2 năm 2011, Membase, Inc ra thông báo về thương vụ sáp nhập với CouchOne (công ty đứng sau thành công của CouchDB) thành một công ty mới mang tên Couchbase, Inc. Đây không chỉ là sáp nhập giữa hai thương hiệu mà còn là sự kết hợp giữa hai nền tảng công nghệ để tạo ra một CouchBase đơn giản, nhanh gọn, linh hoạt trên nền tảng từ CouchDB.

Các mốc quá trình phát triển của CouchBase:

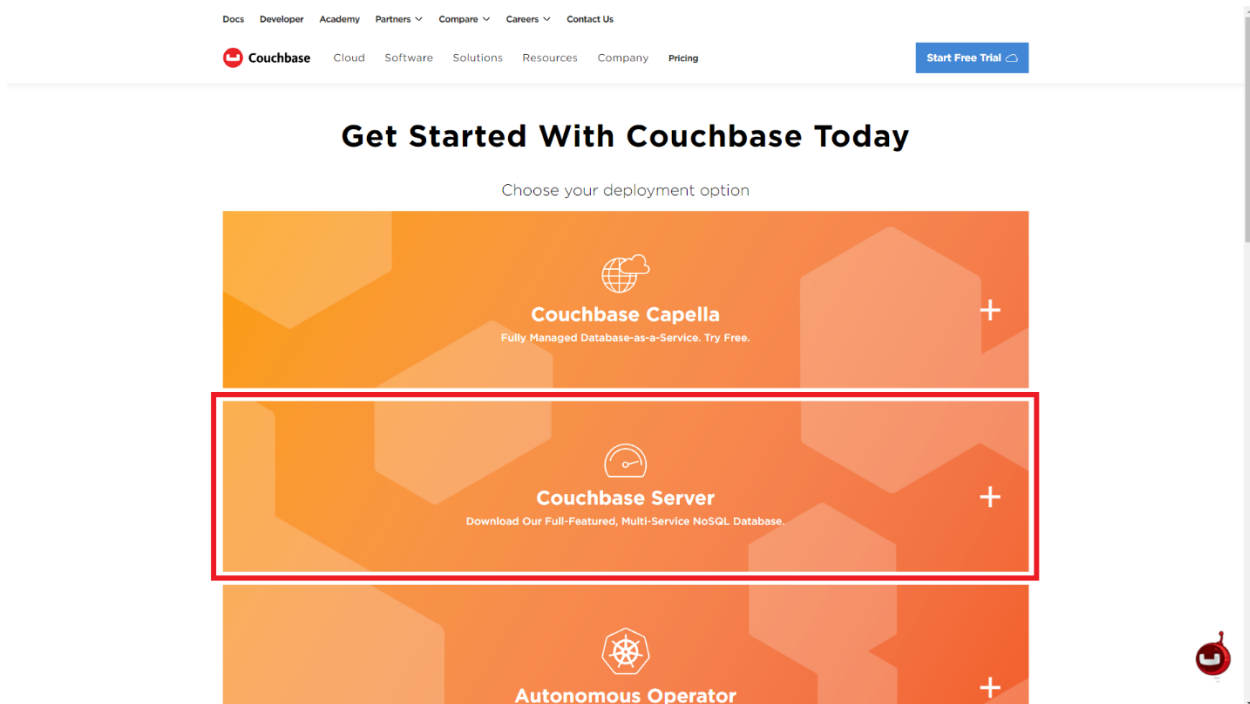
- Năm 2011, chính thức ra mắt sau khi sáp nhập giữa CouchOne và MemBase
- Năm 2012, CouchBase server 2.0 chuyển giao hướng tiếp cận từ csdl key-value sang csdl hướng JSON document.
- Năm 2014, cho ra đời csdl Nosql phiên bản đầu tiên dành cho di động
- Năm 2015, CouchBase Server 4.0 tích hợp thêm truy vấn bằng N1QL và Multi-Dimensional Scaling
- Năm 2017, CouchBase Server 5.0 giới thiệu về Full-Text Search và ephemeral buckets.

- Năm 2018, CouchBase Server 6.0 loại bỏ ETL dưới sự ra đời của SQL++: CouchBase Autonomous Operator cho Kubernetes.
- Năm 2019, CouchBase Server 6.5 tích hợp thêm giao tác phân tán ACID.
- Năm 2020, CouchBase cho ra mắt CouchBase Cloud - sản phẩm Database-as-a-Service.
- Năm 2021, CouchBase được niêm yết giá trên sàn Nasdaq dưới ký hiệu BASE.

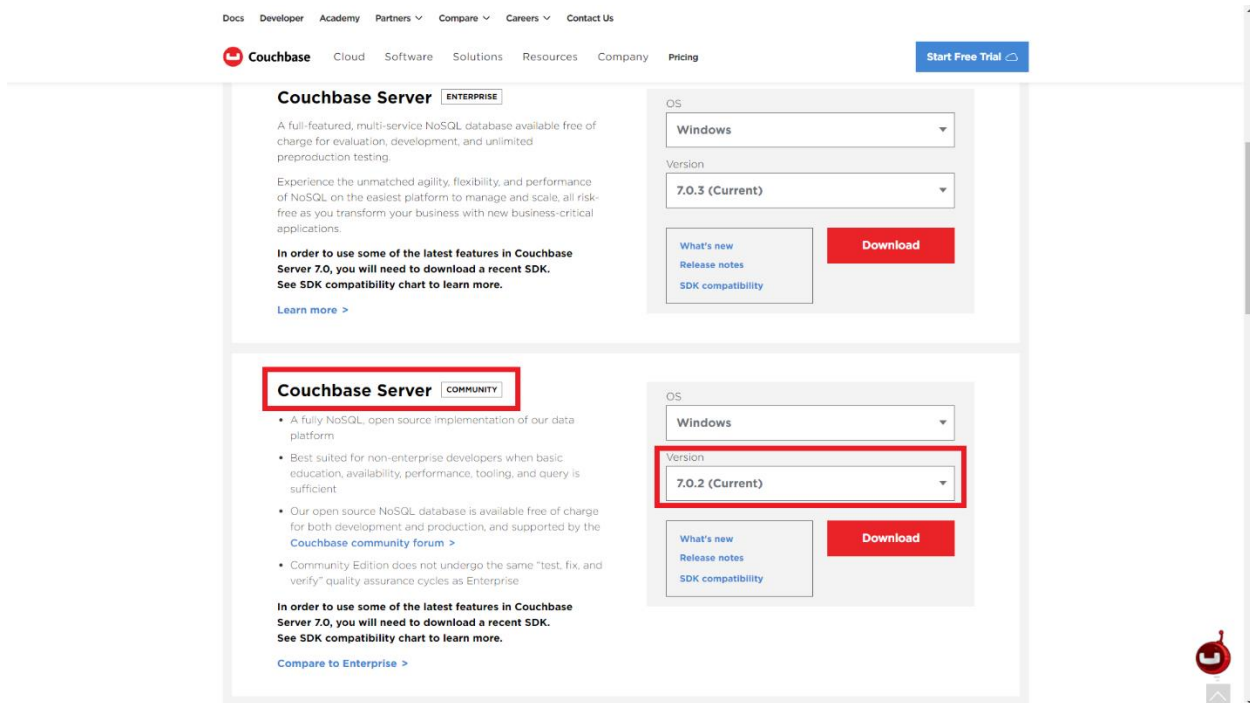
2. Tạo và thêm mới dữ liệu

2.1. Tải Couchbase

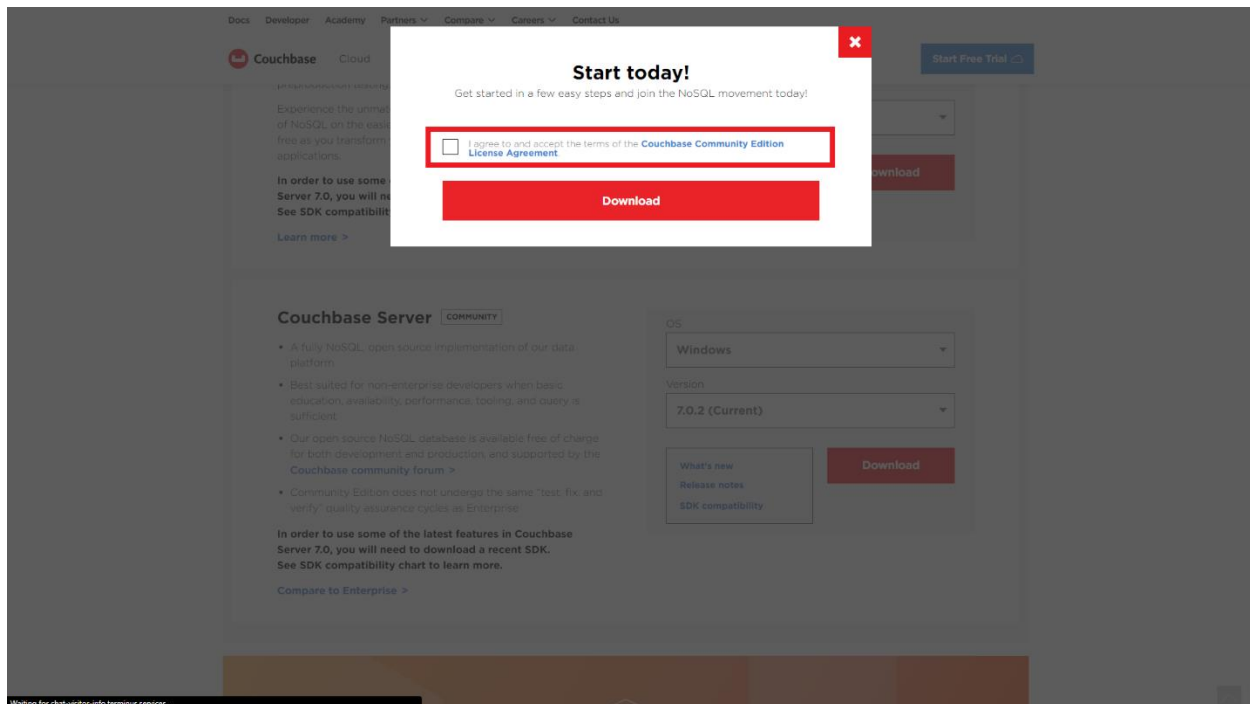
Bước 1. Vào trang web <https://www.couchbase.com/downloads> và chọn banner “Couchbase Server”



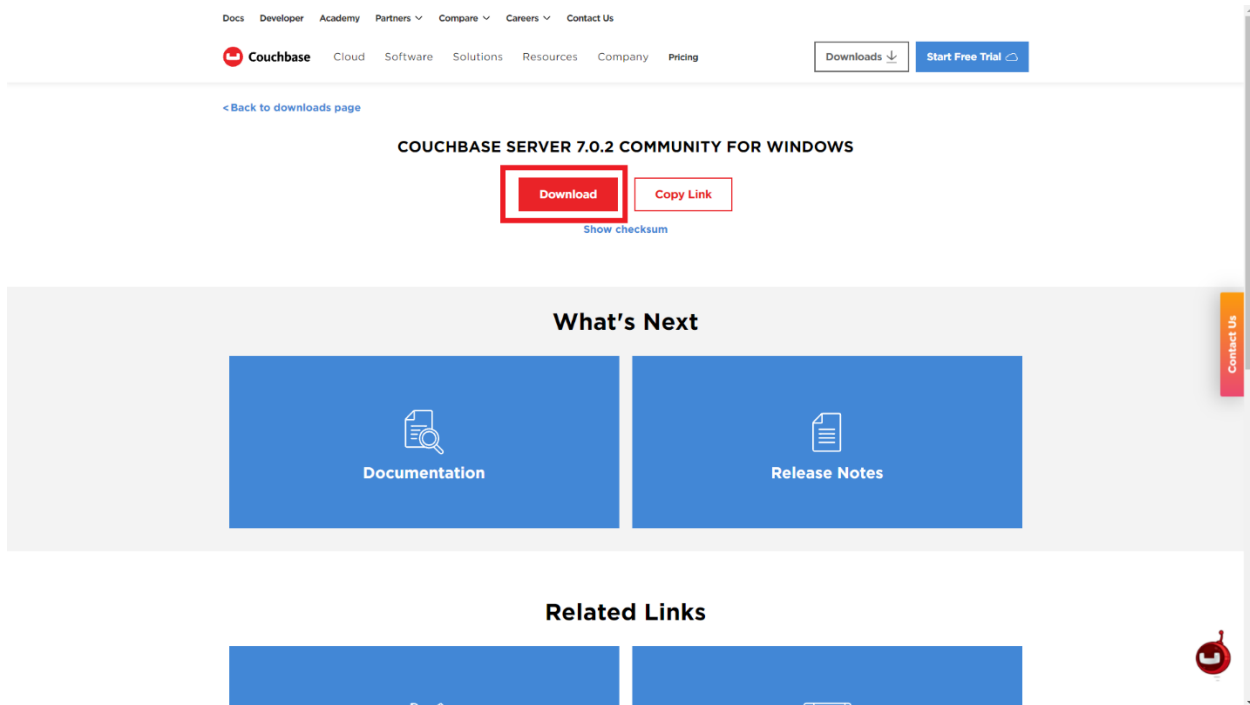
Bước 2. Chọn version của Couchbase Server Community Version và click vào nút “Download”.
Khuyến nghị dùng version mới nhất của Couchbase có tag “Current” ngay sau phiên bản.



Bước 3. Sau khi click nút “Download”, website của Couchbase sẽ hiện bảng chấp thuận điều khoản sử dụng phần mềm. Hãy đọc kỹ và tích vào ô “I agree to and ...”, sau đó nhấn nút “Download”.

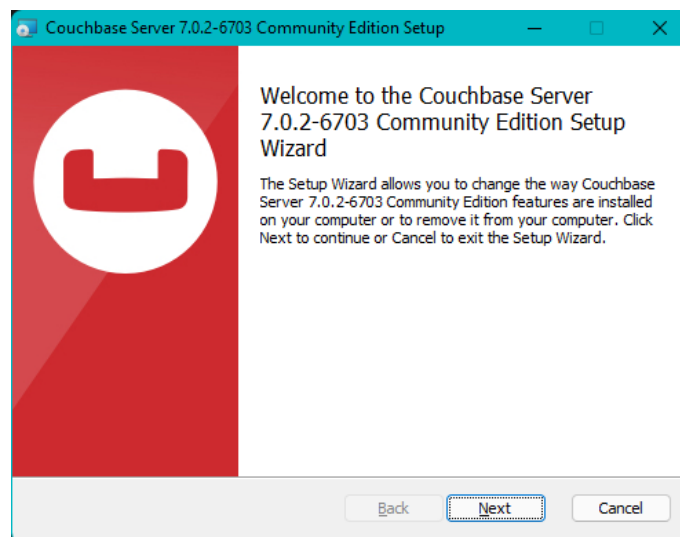


Bước 4. Cuối cùng website sẽ thực hiện điều hướng đến trang web phụ để thực hiện tải về. Click vào button “Download” để thực hiện tải về phần mềm cài đặt Couchbase.

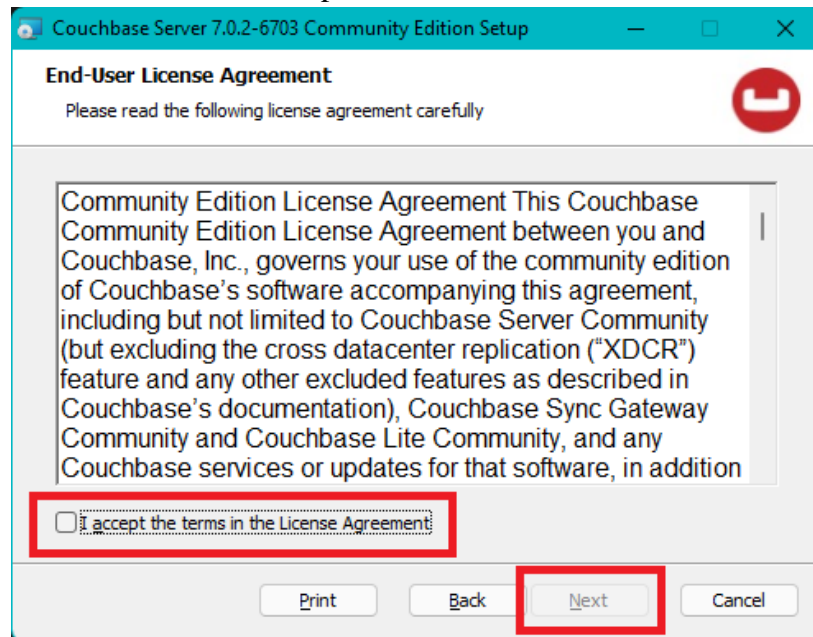


2.2. Thực hiện cài đặt Couchbase

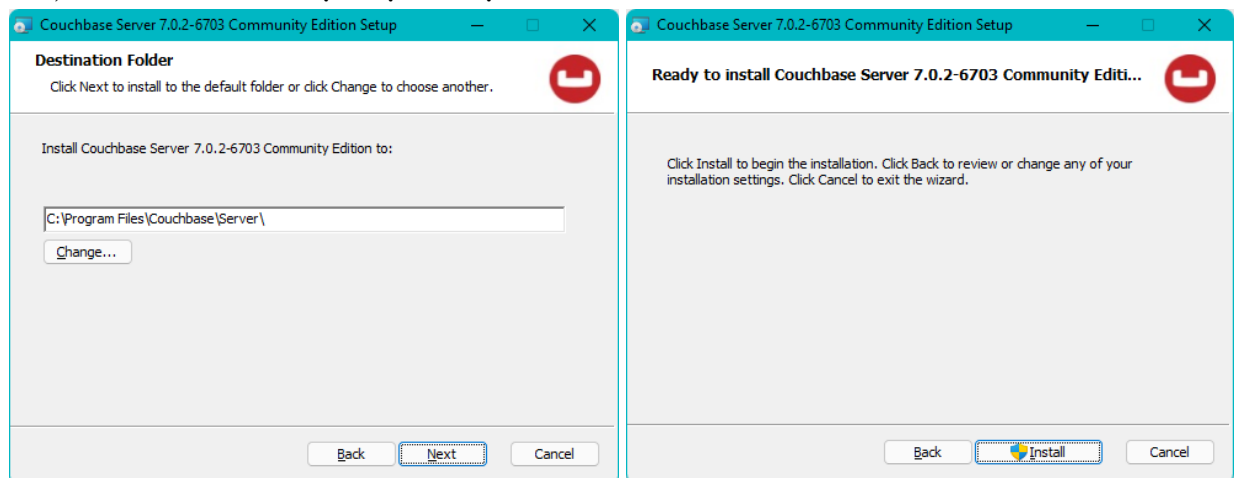
Bước 1. Chạy trình cài đặt Couchbase đã tải về có tên là “couchbase-server-community_7.0.2-windows_amd64.msi”.



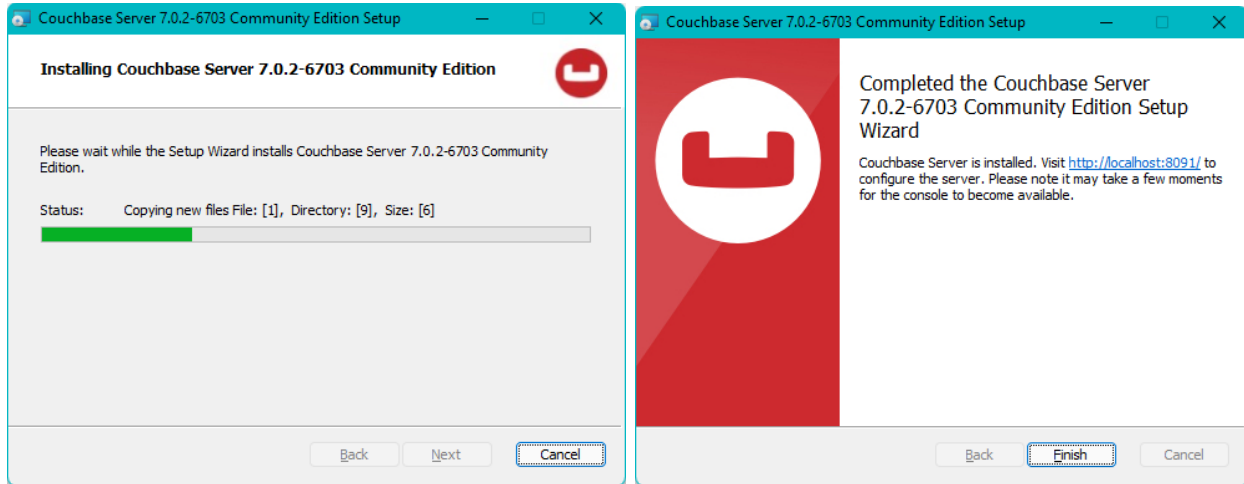
Bước 2. Thực hiện nhấn “Next” để tiếp tục chuyển sang phần Điều khoản sử dụng. Đọc kỹ điều khoản sử dụng và nhấn vào “Next” để tiếp tục.



Bước 3. Chọn đường dẫn cài đặt Couchbase và nhấn “Next” để sang bước tiếp theo. Ở bước tiếp theo, nhấn “Install” để thực hiện cài đặt Couchbase.



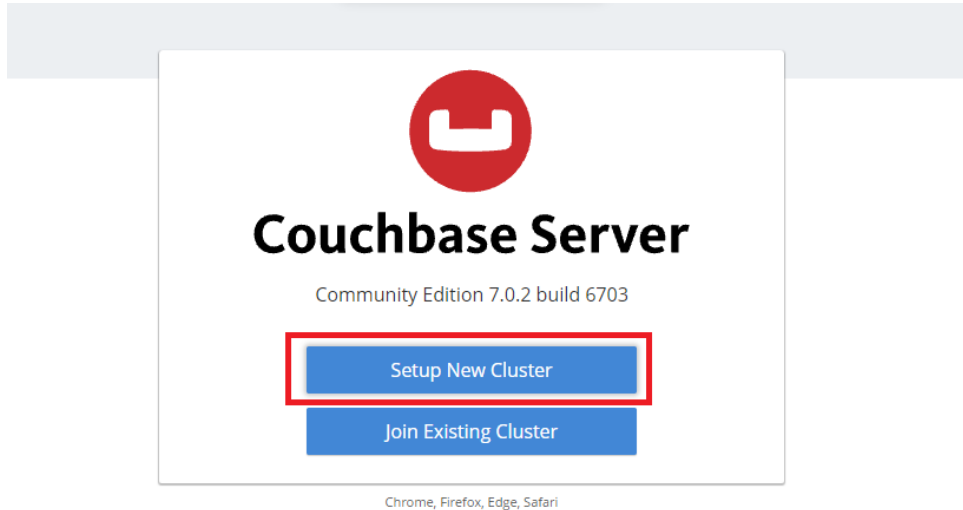
Bước 4. Chờ cho trình cài đặt Couchbase tiến hành cài đặt phần mềm (*quá trình này có thể mất một vài phút*). Và nếu quá trình cài đặt thành công, chúng ta sẽ có thông báo hoàn tất. Khi này, nhấn vào nút “Finish” để kết thúc quá trình cài đặt.



Bước 5. Thực hiện khởi động lại máy để hoàn tất quá trình cài đặt.

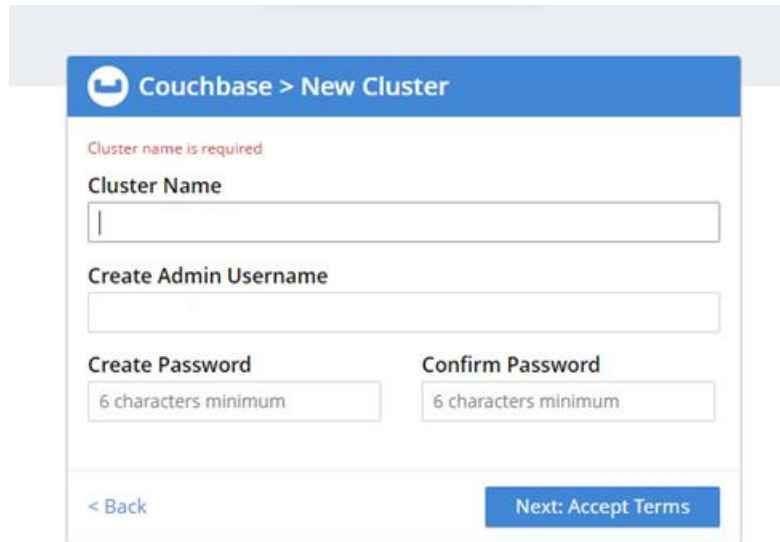
2.3. Thực hiện cài đặt Cluster (cụm) mới

Bước 1. Sau khi cài đặt hoàn tất Couchbase, ta vào đường dẫn <http://localhost:8091/> để vào trình quản lý cơ sở dữ liệu của Couchbase. Và click vào nút “Setup New Cluster” để bắt đầu cài đặt.

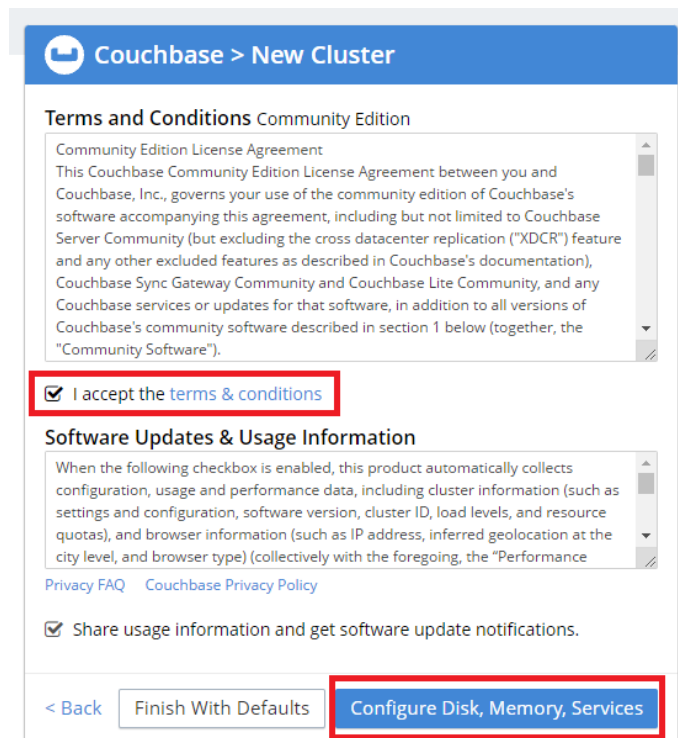


Bước 2. Thực hiện đặt tên cho cluster, username và mật khẩu cho tài khoản quản trị cho cluster đó. Ví dụ:

- **Cluster Name:** IS211 – Zehn
- **Create Admin Username:** master
- **Password:** 123456



Bước 3. Đọc kỹ điều khoản sử dụng và tích vào ô “*I accept the term & conditions*” và click vào nút “Configure Disk, Memory, Services”. Trong trường hợp sử dụng cấu hình mặc định của Couchbase, click vào nút “Finish with defaults”.



Bước 4. Thực hiện đặt địa chỉ IP Address là địa chỉ IP trên Radmin VPN của máy thực hiện cài đặt (trong trường hợp này là 26.167.214.119). Sau đó thực hiện tùy chỉnh các thiết lập bên dưới như “Service Memory Quotas” hoặc “Index Storage Setting”. Cuối cùng là nhấn Save & Finish để hoàn tất quá trình tạo Cluster mới.

Couchbase > New Cluster > Configure

Host Name / IP Address Fully-qualified domain name

26.167.214.119

Service Memory Quotas ^① Per service / per node

<input checked="" type="checkbox"/> Data	3202	MiB
<input checked="" type="checkbox"/> Index	512	MiB
<input checked="" type="checkbox"/> Search	512	MiB
<input checked="" type="checkbox"/> Query	-----	

TOTAL QUOTA 4226MiB

RAM Available 6447MiB Max Allowed Quota 5423MiB

Index Storage Setting

☒ Standard Global Secondary

☐ Memory-Optimized ^①

Data Disk Path Path cannot be changed after setup

c:/Program Files/Couchbase/Server/var/lib/couchbase/data

Free: 60 GiB

Indexes Disk Path Used by GSI, FTS, and Views

c:/Program Files/Couchbase/Server/var/lib/couchbase/data

Free: 60 GiB

Eventing Disk Path Path cannot be changed after setup

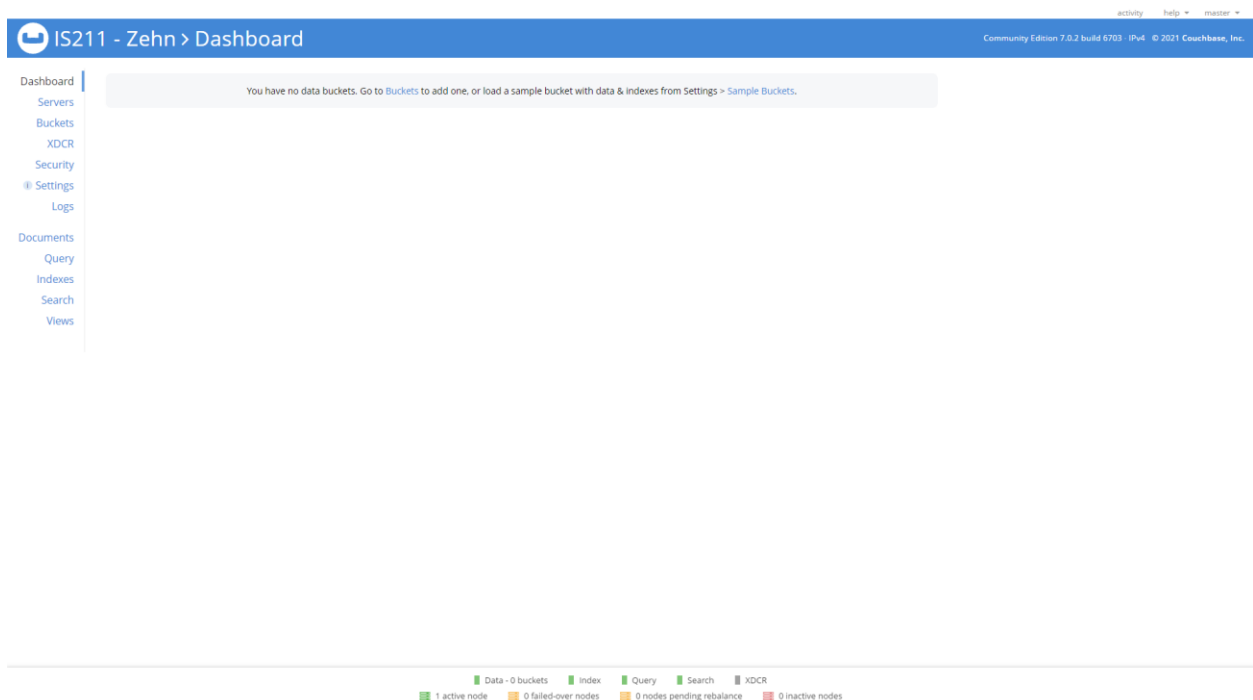
c:/Program Files/Couchbase/Server/var/lib/couchbase/data

Free: 60 GiB

< Back

Save & Finish

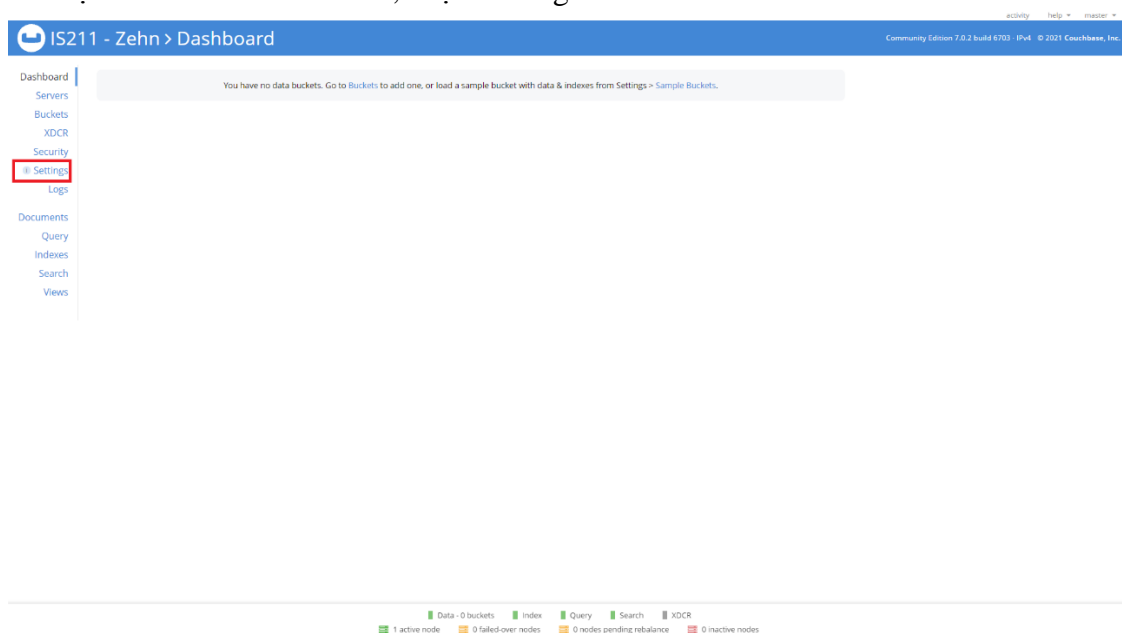
Bước 5. Sau khi tạo thành công, ta sẽ được điều hướng đến trang Dashboard của cluster trên Couchbase.



2.4. Thực hiện thêm dữ liệu

2.4.1. Nạp và sử dụng dữ liệu có sẵn

Bước 1. Tại Dashboard của cluster, chọn Setting



Bước 2. Tại phần Setting của Cluster, chọn tab “Sample Buckets”.

activity help master

IS211 - Zehn > Settings

General Auto-Compaction Alerts **Sample Buckets**

Dashboard
Servers
Buckets
XDCR
Security
Settings
Logs
Documents
Query
Indexes
Search
Views

Cluster Name
IS211 - Zehn

Memory Quotas per server node
Note: Total Couchbase RAM allocation for each node cannot exceed 90% of its available memory.

Data 3202 MIB

Query -----

Index 512 MIB

Search 512 MIB

Current Version
Couchbase Server Community Edition 7.0.2 build 6703

☒ Share usage information with Couchbase and get software update notifications.

Couchbase Server Community Edition 7.0.3-7031 is available. New features and improvements in this release can be found in the [release notes](#).

Node Availability ⓘ
☒ Auto-failover after 120 seconds

► For Ephemeral Buckets

Rebalance Settings
► Advanced Rebalance Settings

Data Settings
► Advanced Data Settings

Query Settings
► Advanced Query Settings

Index Storage Mode
☒ Standard Global Secondary
☐ Memory-Optimized ⓘ

► Advanced Index Settings

XDCR Maximum Processes
4

Save Cancel/Reset

Bước 3. Chọn các bộ Bucket có sẵn của Couchbase và nhấn vào nút “Load Sample Data” để thực hiện nạp thêm dữ liệu mẫu có sẵn để sử dụng.

activity help master

IS211 - Zehn > Settings

General Auto-Compaction Alerts Sample Buckets

Dashboard
Servers
Buckets
XDCR
Security
Settings
Logs
Documents
Query
Indexes
Search
Views

Sample Buckets
Sample buckets contain example data, views, and indexes for your experimentation.

Sample buckets — like all buckets in Couchbase Server 5.0+ — can only be accessed by a user with privileges for that bucket.

Available Samples

☐ beer-sample

☐ gamesim-sample

☒ travel-sample

Load Sample Data

Installed Samples
none

Bước 4. Để kiểm tra dữ liệu đã được nạp đầy đủ hay chưa, ta vào phần “Buckets” của Cluster. Khi thấy có bucket có tên là “travel-sample” như hình bên dưới nghĩa là dữ liệu mẫu đã được nạp thành công.

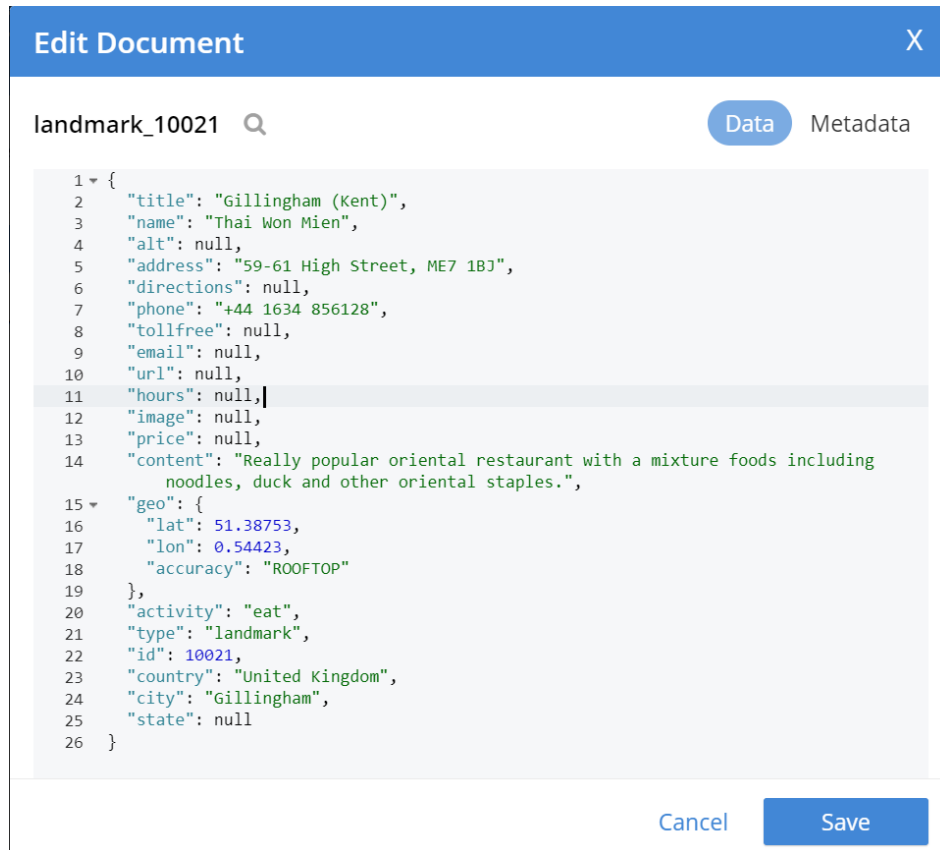
The screenshot displays the 'Buckets' page in the Elasticsearch UI for a cluster named 'IS211 - Zehn'. The page features a blue header with the cluster name and a search bar. On the left, a sidebar lists various navigation options. The main content area shows a table of buckets. A single bucket named 'travel-sample' is listed with the following details: 0 items, 100% resident, 0 ops/sec, 3.87MiB / 200MiB RAM used/quota, and 0B disk used. To the right of the table, there are links for 'Documents' and 'Scopes & Collections'. The top right corner includes links for 'activity', 'help', and 'master'.

name	items	resident	ops/sec	RAM used/quota	disk used
travel-sample	0	100%	0	3.87MiB / 200MiB	0B

2.4.2. Thêm dữ liệu mới

2.4.2.1. Cách tổ chức dữ liệu của Couchbase

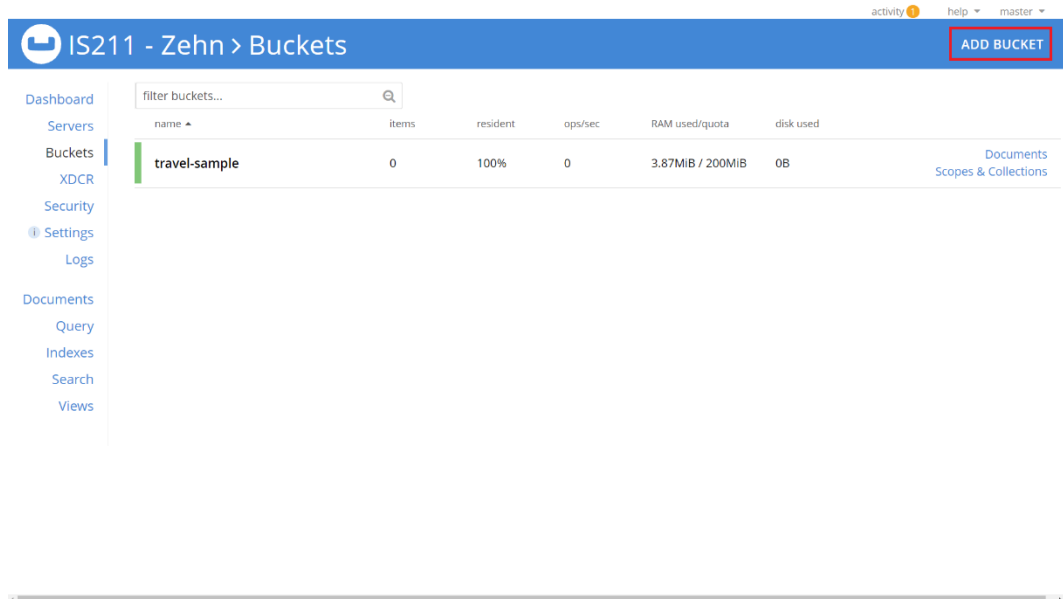
Trong Couchbase, các dữ liệu sẽ được lưu trữ dưới định dạng JSON, có thể hiểu đơn giản là tập các cặp dữ liệu Key-value (ví dụ như hình bên dưới). Các đơn vị lưu trữ bên trong Couchbase được xác định như sau:



- **Document:** Với mỗi một file JSON được lưu bên trong Couchbase, ta có 1 document. Những document này sẽ được quản lý bởi Collection lớn hơn.
- **Collection:** Là tập các document có liên quan với nhau. Mỗi cluster được lưu trữ không quá 1000 collection. Và mỗi collection đều có tên riêng biệt, không trùng với collection khác.
- **Scope:** Là tập các Collection có liên quan với nhau. Mỗi cluster được lưu trữ không quá 1000 scope. Và mỗi scope đều phải có tên riêng biệt, không trùng với scope khác.
- **Bucket:** Là đơn vị quản lý lớn nhất, chứa các Scope, Collection, và Document. Mỗi cluster chứa không quá 30 Bucket.

2.4.2.2. Thực hiện tạo mới dữ liệu bằng giao diện Couchbase

Bước 1. Tại Dashboard của cluster, chọn vào phần Bucket, khi này ta sẽ click vào nút “Add Bucket”.



Bước 2. Khi này chúng ta đặt tên cho Bucket và lựa chọn Bucket Type (mặc định là “Couchbase”). Và sau đó nhấn “Add Bucket”.

Add Data BucketX

Name

Bucket name cannot be empty

Memory Quota

in megabytes per server node

3002MiB

other buckets (200MiB)this bucket (2.93GiB)available (0B)

Bucket Type

☒ Couchbase

☐ Memcached

☐ Ephemeral

▶ Advanced bucket settings

Cancel

Add Bucket

Bước 3. Khi này, bucket mới đã được tạo ra. Chúng ta sẽ click vào dòng “Scope & Collection” để bắt đầu thêm mới Scope và Collection.

IS211 - Zehn > Buckets

filter buckets...

name	items	resident	ops/sec	RAM used/quota	disk used	
travel-sample	63,288	100%	0	119MiB / 200MiB	62.3MiB	Documents Scopes & Collections
zehn	0	100%	0	33.5MiB / 2.93GiB	8.06MiB	Documents Scopes & Collections

Bước 4. Tại phần quản lý Scope & Collection của Bucket vừa tạo (ví dụ ở đây là “zehn”). Ta click vào “Add Scope”.

IS211 - Zehn > Buckets > Scopes & Collections

zehn

filter scopes

scope name	collections	items	memory used	disk utilization	ops/sec	
_default	1	0	0B	0B	0	Documents Add Collection

< prev 1 next >

Bước 5. Thực hiện đặt tên cho Scope và click vào “Save”. Khi này một scope mới sẽ được tạo ra.

Add ScopeX

Bucket Name

zehn

New Scope Name

zehn-scope

Cancel

Save

Bước 6. Tại phần quản lý Scope & Collection, ta chọn vào dòng “Add Collection” của Scope vừa tạo (ví dụ ở đây là “zehn-scope”).

activityhelpmaster

IS211 - Zehn > Buckets > Scopes & CollectionsADD SCOPE

Dashboard

Servers

Buckets

XDCR

Security

Settings

Logs

Documents

Query

Indexes

Search

Views

zehn

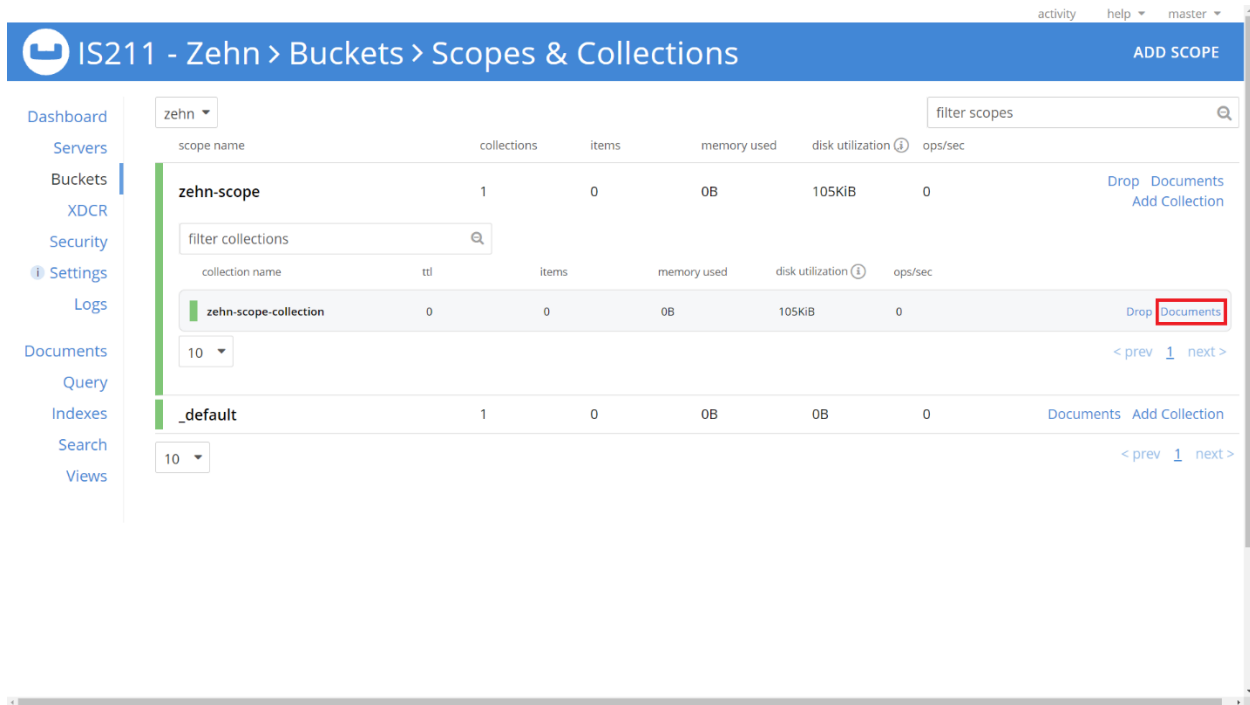
filter scopes

scope name	collections	items	memory used	disk utilization	ops/sec	
zehn-scope	0	-	-	-	-	Drop Documents Add Collection
_default	1	0	0B	0B	0	Documents Add Collection

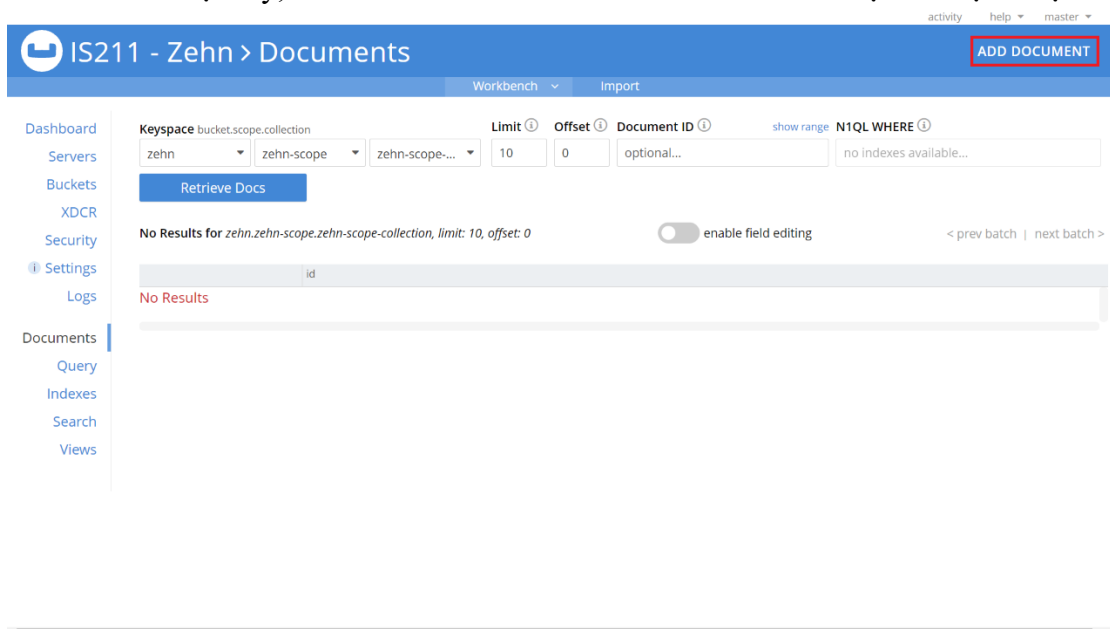
10

< prev 1 next >

Bước 7. Thực hiện đặt tên cho collection và click “Save”. Khi này, một collection mới sẽ được tạo ra và thuộc quyền quản lý của Scope chứa nó. Để thêm một document mới vào collection vừa tạo (ví dụ ở đây là “zahn-scope-collection”), ta click vào chữ “Document” thuộc dòng của collection vừa tạo.




Bước 8. Couchbase sẽ điều hướng giao diện đến phần quản lý Document của collection “zahn-scope-collection”. Tại đây, ta nhấn vào nút “Add Document” để thêm một đơn vị dữ liệu mới.



Bước 9. Thực hiện đặt ID cho Document và thêm giá trị cho Document ấy. Như ví dụ hình bên dưới là thông tin định dạng JSON của một sinh viên. Sau khi hoàn tất, nhấn vào nút “Save” để lưu lại thông tin của Document này.

Create New Document

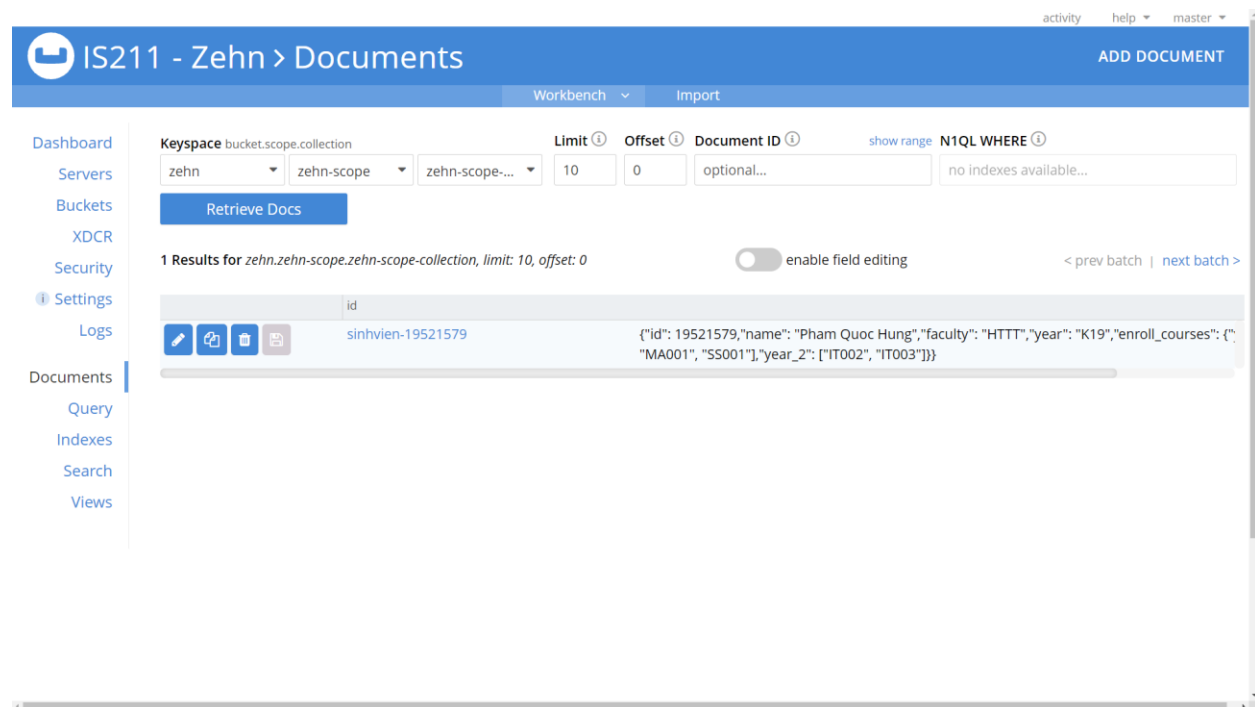
X

ID 

```
1 {  
2   "id": 19521579,  
3   "name": "Pham Quoc Hung",  
4   "faculty": "HTTT",  
5   "year": "K19",  
6   "enroll_courses": {  
7     "year_1": ["IT001", "MA001", "SS001"],  
8     "year_2": ["IT002", "IT003"]  
9   }  
10 }
```

Cancel Save

Bước 10. Khi này, một document mới đã được tạo ra và thuộc collection “zehn-scope-collection”.



2.4.2.3. Thực hiện tạo mới dữ liệu trên Couchbase bằng Python

Bước 1. Để thực hiện thao tác dữ liệu với Couchbase bằng Python, ta cần cài đặt module “Couchbase for Python” bằng lệnh sau trên Terminal (đối với Linux và MacOS) hoặc trên Windows Terminal/PowerShell/Command Prompt (đối với Windows)

```
$ pip install couchbase
```

Bước 2. Thực hiện tạo sẵn Bucket, Scope và Collection bằng giao diện của Couchbase như phần hướng dẫn bên trên.

- Lý do là vì Couchbase giới hạn việc quản lý bucket phải được làm bằng giao diện của Couchbase, thao tác tạo Scope và collection chỉ có thể được thực hiện thông qua Couchbase-CLI hoặc RestAPI của Couchbase.

Bước 3. Thực hiện nhập đoạn code bên dưới vào để bắt đầu thực hiện thao tác dữ liệu với Couchbase.

- 3 Dòng import đầu tiên là những Class cơ bản để có thể tạo connection giữa script Python và Couchbase trên máy local.
- Với tham số đầu tiên của class Cluster được khởi tạo, ta điền vào địa chỉ IP là địa chỉ IP của máy hiện tại trên Radmin VPN. Trong trường hợp không sử dụng Radmin VPN. Ta thay địa chỉ IP đó bằng “localhost”.
- Tham số thứ hai là đối tượng PasswordAuthenticator với hai tham số là tài khoản và mật khẩu của máy chủ dùng để xác thực tài khoản và tạo connection

```

from couchbase.cluster import Cluster, ClusterOptions
from couchbase.cluster import QueryOptions
from couchbase.auth import PasswordAuthenticator

cluster = Cluster('couchbase://26.167.214.119',
                  ClusterOptions(
                      PasswordAuthenticator('master', '123456')
                  )
)

```

Bước 4. Truy cập vào Collection vừa tạo. (ví dụ ở đây là collection “zehn-scope-collection” thuộc scope “zehn-scope” của bucket “zehn”)

```

collection = cluster.bucket('zehn')
                .scope('zehn-scope')
                .collection('zehn-scope-collection')

```

Bước 5. Tạo sẵn một document để lưu vào collection vừa truy cập

```

document = {
    "key1": "value1",
    "key2": 2,
    "key3": [3, 4, 5]
}

```

Bước 6. Thực hiện lưu document với ID là “new-document” vào collection “zehn-scope-collection”.

```

result = collection.insert("new-document", document)

print(result.cas)

```

Kết quả

```

[Running] python -u "d:\zehn.py"
<couchbase.result.MutationResult object at 0x0000000003C41AF0>

[Done] exited with code=0 in 0.555 seconds

```

Bước 7. Kiểm tra lại trên hệ thống, ta thấy document “new-document” đã được cập nhật vào bên trong collection “zehn-scope-collection”. Nghĩa là quá trình thêm mới dữ liệu đã diễn ra thành công

The screenshot shows the 'IS211 - Zehn > Documents' interface. The top navigation bar includes 'activity', 'help', and 'master'. The main header has 'IS211 - Zehn > Documents' and an 'ADD DOCUMENT' button. Below the header, there are tabs for 'Workbench' and 'Import'. The left sidebar contains a navigation menu with 'Dashboard', 'Servers', 'Buckets', 'XDCR', 'Security', 'Settings', 'Logs', 'Documents', 'Query', 'Indexes', 'Search', and 'Views'. The main content area displays a table of documents. The table has columns for 'id' and 'Document ID'. The first document is 'new-document' with a 'Document ID' of 'optional...'. The second document is 'sinhvien-19521579' with a 'Document ID' of 'optional...'. The 'new-document' row is highlighted with a red box. The table also shows a 'Retrieve Docs' button and a 'Limit' of 10. The 'N1QL WHERE' clause is 'no indexes available...'. The 'enable field editing' toggle is turned off. The 'prev batch' and 'next batch' links are visible.

id	Document ID
new-document	optional...
sinhvien-19521579	optional...

Yêu cầu 2, 3:

Cơ chế truy vấn và thao tác dữ liệu phân tán

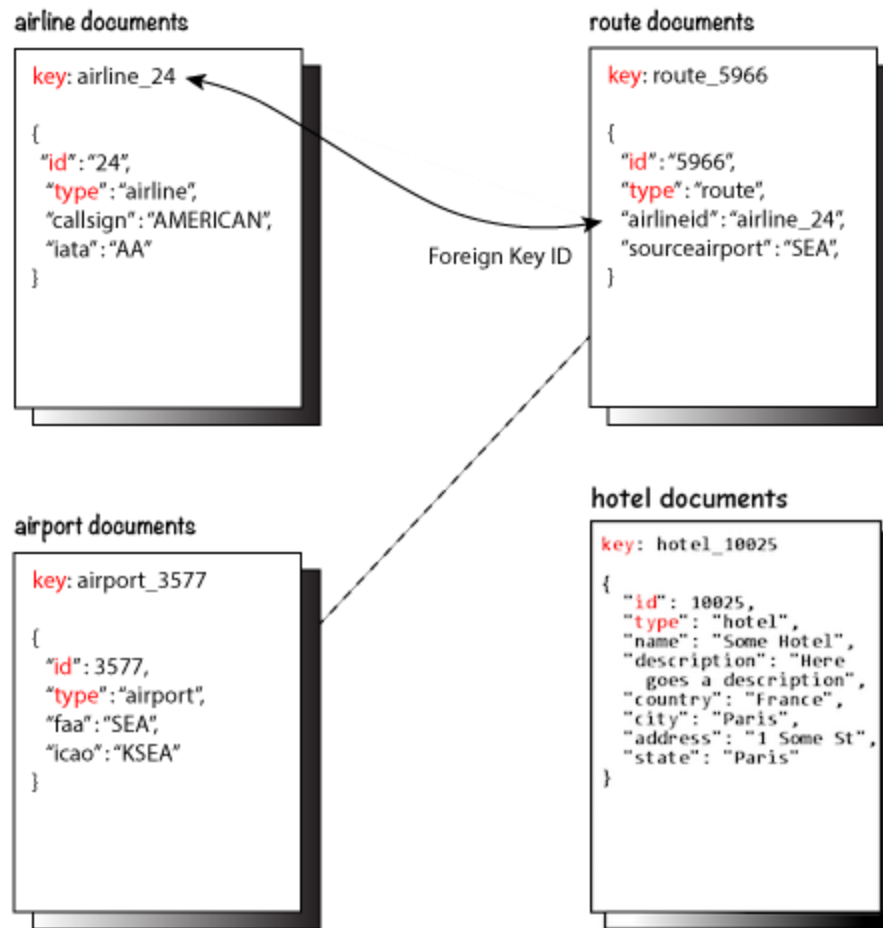
1. Mô tả dữ liệu mẫu: Travel App

Dữ liệu mẫu của Couchbase mà báo cáo này sử dụng có tên là Travel App được lưu trữ trong bucket “travel-sample”. Bucket này bao gồm 6 scope, mỗi scope sẽ có các collection như sau:

STT	Tên Scope	Tên Collection	Số lượng Document
1	inventory	airline	187
2		airport	1968
3		hotel	917
4		landmark	4495
5		route	24024
6	tenant_agent_00	bookings	0
7		users	2
8	tenant_agent_01	bookings	0
9		users	11
10	tenant_agent_02	bookings	0
11		users	20
12	tenant_agent_03	bookings	0
13		users	33
14	tenant_agent_04	bookings	0
15		users	40
16	default	default	31591

Mô hình dữ liệu cho mỗi loại document bao gồm:

- **key**: khoá chính
- **id**: định danh document
- **type**: xác định loại document

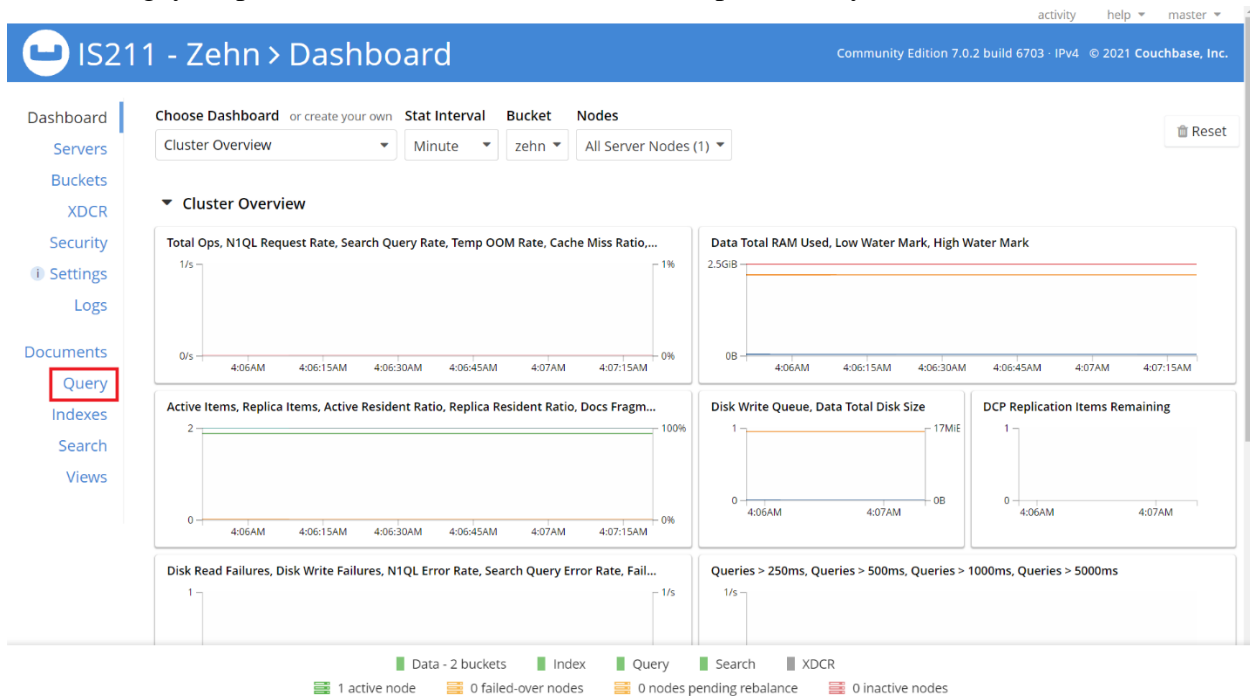


Hình 2. Mối quan hệ giữa các document khác nhau của dữ liệu Travel App.
Thông tin hiển thị bao gồm khoá chính, id, loại cùng với một số trường đại diện của mỗi loại document.

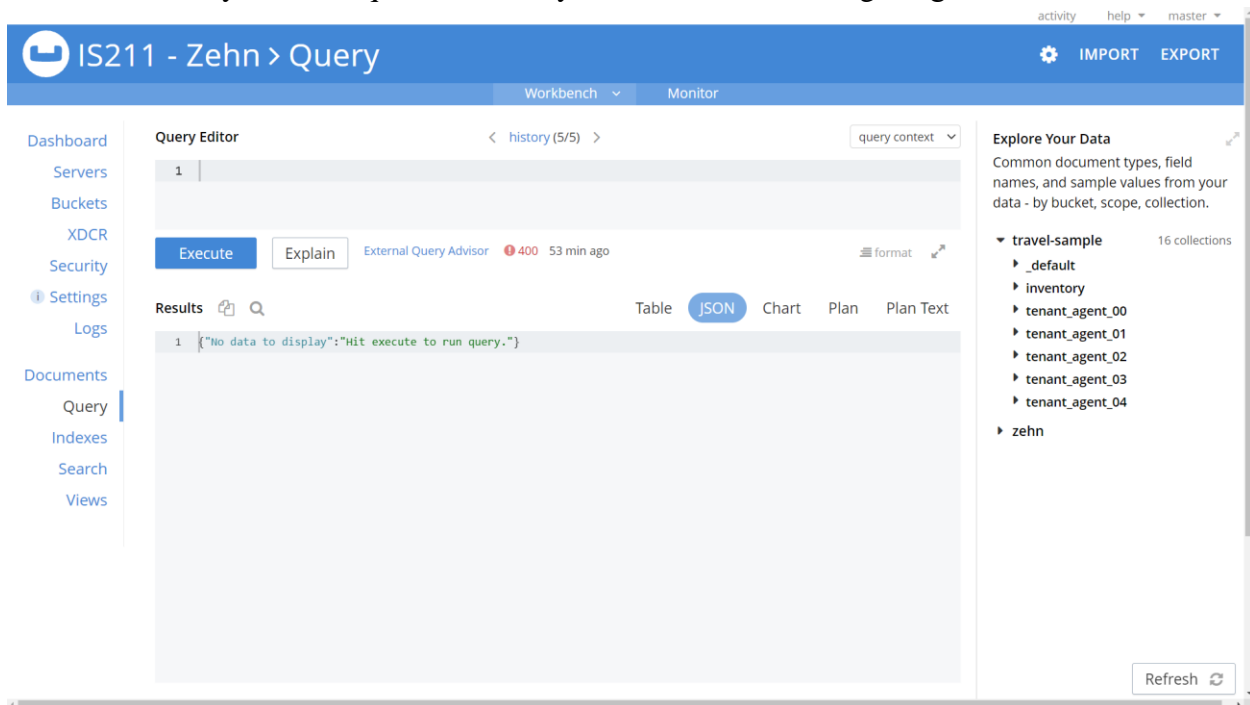
2. Truy vấn dữ liệu

2.1. Truy vấn dữ liệu trực tiếp bằng Query Workbench của Couchbase dùng N1QL Query

Bước 1. Ngay tại phần Dashboard của Cluster, ta chọn phần Query



Bước 2. Bắt đầu viết câu truy vấn bằng ngôn ngữ N1QL tại Query Editor và nhấn “Execute” để thực hiện câu truy vấn. Kết quả của câu truy vấn sẽ được in ra trong bảng “Result”.



Ví dụ: Thực hiện truy vấn danh sách các mã hãng hàng không, quốc gia của hãng hàng không và tên của hãng hàng không đó từ Collection “airline” của Scope “inventory” trong bucket “travel-sample”.

- Trước tiên, ta thực hiện thay đổi Query Context sang “travel-sample” và thay scope là “inventory”.
- Thực hiện câu truy vấn:

```
select a.id, a.name, a.country from airline a
```

- Kết quả câu truy vấn:


Results  

Table JSON Chart Plan Plan Text

```

1 [
2   {
3     "country": "United States",
4     "id": 10,
5     "name": "40-Mile Air"
6   },
7   {
8     "country": "United States",
9     "id": 10123,
10    "name": "Texas Wings"
11  },
12  {
13    "country": "United States",
14    "id": 10226,
15    "name": "Atifly"
16  },
17 ]

```

- Kết quả câu truy vấn dưới dạng bảng:


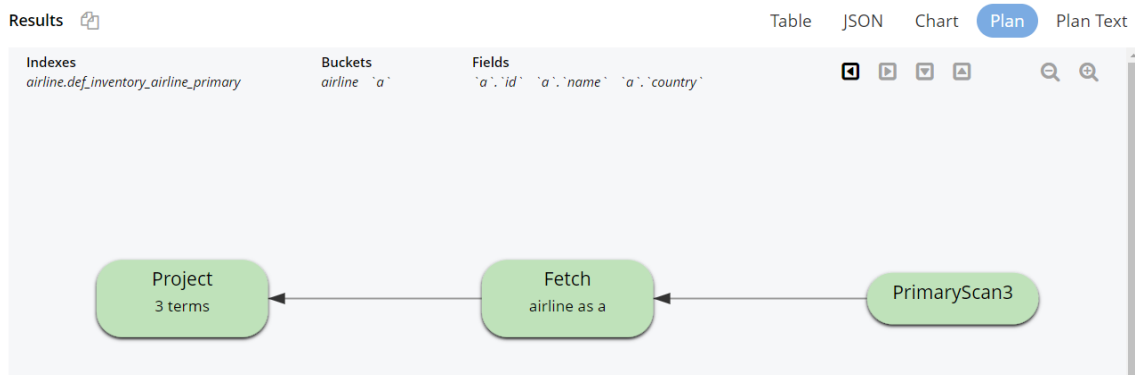
Results 

Table JSON Chart Plan Plan Text

country	id	name
United States	10	40-Mile Air
United States	10123	Texas Wings
United States	10226	Atifly
United Kingdom	10642	Jc royal.britannica
United States	10748	Locair
United States	10765	SeaPort Airlines
United States	109	Alaska Central Express
United Kingdom	112	Astraeus
France	1191	Air Austral
France	1203	Airlinair
United States	1316	AirTran Airways
United States	13391	U.S. Air
United Kingdom	1355	British Airways

- Explain Plan của câu truy vấn:



2.2. Truy vấn dữ liệu trực tiếp bằng Python không dùng NIQL Query

Để lấy được dữ liệu, trước hết ta cần xác định bucket, scope và collection chứa document cần lấy. Để xác định, ta thực hiện tương tự **bước 2.4** của **Yêu cầu 1 phần 2.4.2**. Trong ví dụ bên dưới, ta sẽ dùng đoạn mã như sau để xác định collection chứa document nằm trong dữ liệu mẫu.

```
collection = cluster.bucket('travel-sample')
                .scope('inventory')
                .collection('airport')
```

Để có thể lấy document bằng ID của document đó, ta dùng phương thức `get()` của lớp `couchbase.collection.Collection`:

<code>get(key, *option, **kwargs)</code>		
Tham số	Kiểu dữ liệu	Ý nghĩa
<code>key</code>	Chuỗi	Key (hay ID) của document được lưu trong collection
Kiểu trả về	Đối tượng thuộc lớp <code>couchbase.result.GetResult</code> Dùng để lưu trữ kết quả truy vấn dữ liệu.	

Vì phương thức `get()` trả về một đối tượng thuộc lớp `GetResult`, thế nên ta cần dùng thuộc tính `content_as` của lớp `couchbase.result.GetResult` để chuyển đổi kiểu dữ liệu.

<code>content_as[datatype]</code>		
Tham số	Kiểu dữ liệu	Ý nghĩa
<code>datatype</code>	Kiểu dữ liệu	Xác định kiểu dữ liệu đích mà document sẽ được chuyển đổi
Kiểu trả về	Tuỳ thuộc vào kiểu dữ liệu của <code>datatype</code> mà kết quả trả về sẽ khác nhau	

Ví dụ:

```
import couchbase
from couchbase.cluster import Cluster, ClusterOptions
from couchbase.cluster import QueryOptions
from couchbase.auth import PasswordAuthenticator

cluster = Cluster('couchbase://26.167.214.119',
                  ClusterOptions(
                      PasswordAuthenticator('master', '123456')
                  ))

collection = cluster.bucket('travel-sample')
                .scope('inventory')
                .collection('airport')

result = collection.get('airport_1257')

print(result.content_as[str])
```

Kết quả

```
[Running] python -u "d:\zehn.py"
{'id': 1257, 'type': 'airport', 'airportname': 'Couterne',
'city': "Bagnole-de-l'orne", 'country': 'France', 'faa': None,
'icao': 'LFA0', 'tz': 'Europe/Paris', 'geo': {'lat': 48.545836,
'lon': -0.387444, 'alt': 718.0}}

[Done] exited with code=0 in 0.369 seconds
```

2.3. Truy vấn dữ liệu bằng Python dùng N1QL Query

Để truy vấn dữ liệu dùng N1QL query, ta có thể dùng phương thức `query()` của lớp `couchbase.cluster.Cluster`.

query(statement, *option, **kwargs)		
Tham số	Kiểu dữ liệu	Ý nghĩa
statement	Chuỗi	Chứa câu truy vấn N1QL
Kiểu trả về	Đối tượng thuộc lớp <code>couchbase.result.QueryResult</code> Dùng để lưu trữ kết quả truy vấn dữ liệu.	

- Lưu ý:

- Khi sử dụng phương thức này, trong câu truy vấn ta cần xác định rõ collection, scope và bucket theo quy tắc sau. Trong đó, tên bucket phải được đặt trong hai dấu “huyền”:

``bucket`.scope.collection`

- Đối với trường hợp sử dụng câu truy vấn trực tiếp (*không tham số*) thì ta có thể viết thẳng câu truy vấn vào tham số statement.

Ví dụ:

```
result = cluster.query(
    'select * from airport where airport.id = 10'
)
```

- Đối với trường hợp sử dụng câu truy vấn có tham số, ta đặt các tham số bên trong chuỗi chứa câu truy vấn. Với trường hợp sử dụng positional parameter (*tạm dịch: Tham số có thứ tự*) thì ta sử dụng kí hiệu **\$n** với **n** là vị trí của các tham số từ trái sang phải bắt đầu từ 1. Ngược lại, với named parameter (*tạm dịch: tham số có định danh*) thì ta sử dụng kí hiệu **\$x** với **x** là tên của tham số đó.

Ví dụ:

```
result = cluster.query(
    "SELECT ts.*
    FROM `travel-sample`.inventory.airport
    WHERE city = $1",
    "San Jose"
)
```

```
result = cluster.query(
    "SELECT ts.*
    FROM `travel-sample`.inventory.airport
    WHERE city = $city",
    city='San Jose'
)
```

Sau khi đã có kết quả truy xuất, ta dùng vòng lặp for-each với đối tượng `couchbase.result.QueryResult` nhận được, với mỗi row của đối tượng đó là một document thoả điều kiện truy vấn của câu truy vấn.

Ví dụ:

```
import couchbase
from couchbase.cluster import Cluster, ClusterOptions
from couchbase.cluster import QueryOptions
from couchbase.auth import PasswordAuthenticator

cluster = Cluster('couchbase://26.167.214.119',
                  ClusterOptions(
                      PasswordAuthenticator('master', '123456')
                  ))

result = cluster.query(
    "SELECT *
    FROM `travel-sample`.inventory.airport
    WHERE airport.city = 'Los Angeles'"
)

for row in result:
    print(row)
```

Kết quả

```
[Running] python -u "d:\zahn.py"
{'airport': {'id': 9533, 'type': 'airport', 'airportname': 'Chatsworth Station', 'city': 'Chatsworth',
'country': 'United States', 'faa': 'CWT', 'icao': None, 'tz': 'America/Los_Angeles', 'geo': {'lat': 34.
256944, 'lon': -118.598889, 'alt': 978.0}}}
{'airport': {'id': 9540, 'type': 'airport', 'airportname': 'Deer Harbor Seaplane', 'city': 'Deer
Harbor', 'country': 'United States', 'faa': 'DHB', 'icao': None, 'tz': 'America/Los_Angeles', 'geo':
{'lat': 48.618397, 'lon': -123.00596, 'alt': 0.0}}}
{'airport': {'id': 9541, 'type': 'airport', 'airportname': 'San Diego Old Town Transit Center',
'city': 'San Diego', 'country': 'United States', 'faa': 'OLT', 'icao': None, 'tz': 'America/
Los_Angeles', 'geo': {'lat': 32.7552, 'lon': -117.1995, 'alt': 0.0}}}

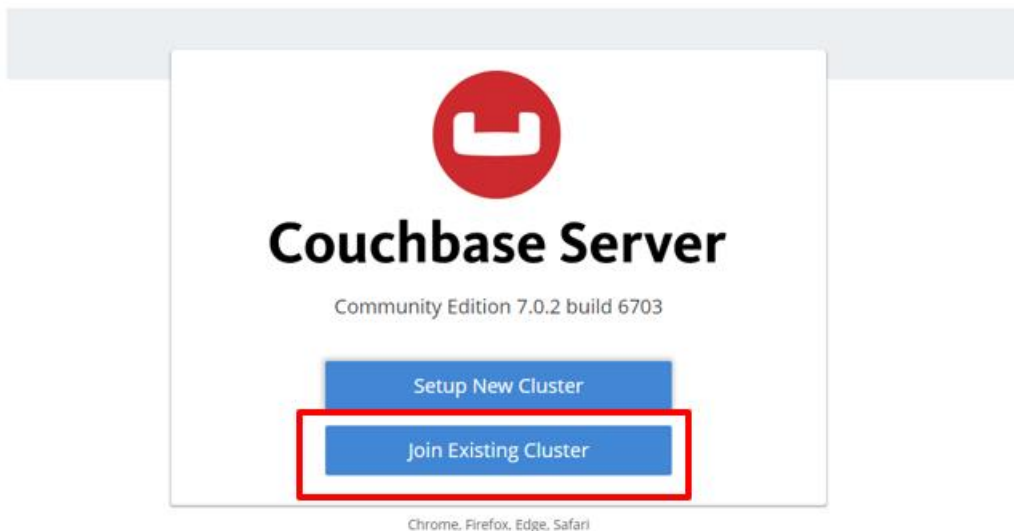
[Done] exited with code=0 in 0.483 seconds
```


3. Cài đặt trên nhiều máy


Bước 1. Trên mỗi máy, ta thực hiện tải phần mềm cài đặt Couchbase server installer tương tự như ở *Yêu cầu 1, phần 2.1*. Và thực hiện cài đặt tương tự như ở *Yêu cầu 1, phần 2.2*.

Bước 2. Với máy thứ nhất, tạm gọi là **máy chủ**. Thực hiện cài đặt Cluster và nạp dữ liệu mẫu Travel App (*bucket “travel-sample”*) theo *Yêu cầu 1, phần 2.3*.

Bước 3. Với các máy tiếp theo, tạm gọi là **các máy trạm**. Tại màn hình đăng nhập tương tự như tại *Yêu cầu 1, phần 2.3*, **Bước 1 phần 2.3**, thay vì chọn “Create new cluster”, ta chọn “Join existing cluster”.



Bước 4. Điền “Cluster Host Name/IP Address”, “Create Admin Username”, “Cluster Admin Password” của máy chủ đã cài đặt (xem lại Yêu cầu 1, phần 2.3, Bước 2). Tiếp theo, điền “Host Name/ IP Address” là địa chỉ IP của máy trạm này trên Radmin VPN (trong trường hợp này là 26.46.70.3)

 Couchbase > Join Cluster

Cluster Host Name/IP Address

26.167.214.119

Create Admin Username

master

Cluster Admin Password

.....

▼ Configure Services & Settings For This Node

Host Name / IP Address

Fully-qualified domain name

26.46.70.3

☒ Data

☒ Index

☒ Search

☒ Query

Data Disk Path

Path cannot be changed after setup

c:/Program Files/Couchbase/Server/var/lib/couchbase/data

Free: 40 GiB

Indexes Disk Path

Used by GSI, FTS, and Views

c:/Program Files/Couchbase/Server/var/lib/couchbase/data

Free: 40 GiB

Eventing Disk Path

Path cannot be changed after setup

c:/Program Files/Couchbase/Server/var/lib/couchbase/data

Free: 40 GiB

< Back

Join Cluster

Bước 5. Nhấn nút “Join Cluster”. Khi hiển thị Dashboard của cluster (*IS211 – Zehn*) là đã đăng ký truy cập được cluster của máy chủ thành công.

Bước 6. Tại máy chủ, khi một máy trạm đã đăng ký truy cập thì sẽ xuất hiện thông báo như sau. Điều này đồng nghĩa máy trạm vẫn chưa được phép tiếp cận dữ liệu. Để giải quyết vấn đề này, ta click vào nút “Rebalance” để cập traffic cho máy trạm đã đăng ký.

The screenshot shows the Couchbase dashboard for cluster 'IS211 - Zehn'. In the top right corner, there is a 'Rebalance' button highlighted with a red rectangle. Below the navigation menu, a table lists servers. The second server, '26.46.70.3', is highlighted with a red rectangle. Its status is 'New node | Not available for traffic | REBALANCE to finish adding node'. A 'Cancel Add' button is visible next to it.

name	services	CPU	RAM	swap	disk used	items
26.167.214.119	data query index search	0.0%	---	---	---	0/0
26.46.70.3	data query index search	5.1%	51.2%	42.9%	---	0/0

Bước 7. Sau khi được cập traffic để truy cập dữ liệu, tại mục Servers của Couchbase dashboard, ta sẽ thấy trạng thái máy trạm sẽ có thông tin như sau

The screenshot shows the detailed view of server '26.46.70.3' in the Couchbase dashboard. The 'Stop Rebalance' button is highlighted with a red rectangle. The server's status is now '40/310' items. Below the server information, there are sections for 'Memory' and 'Disk Usage' with progress bars showing usage and remaining capacity.

name	services	CPU	RAM	swap	disk used	items
26.167.214.119	data query index search	0.0%	---	---	---	0/0
26.46.70.3	data query index search	6.6%	52.9%	45.8%	640KiB	40/310

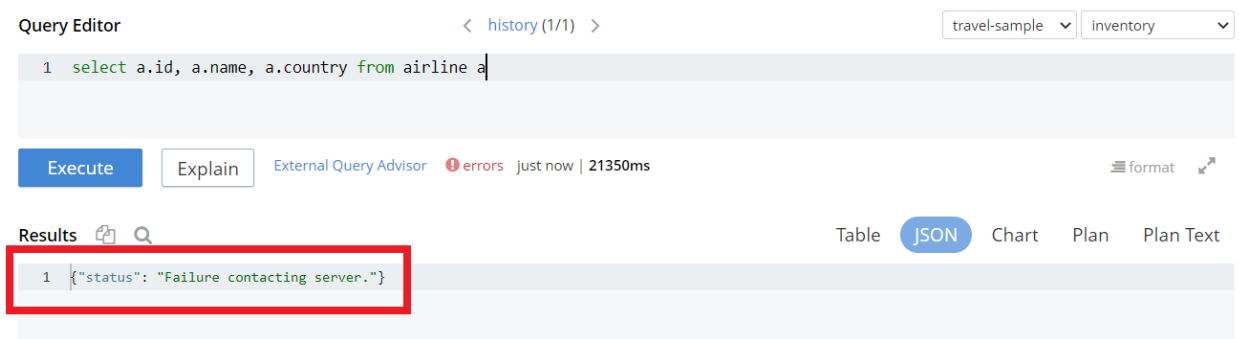
4. Lấy dữ liệu qua lại giữa hai máy bằng NoSQL

4.1. Yêu cầu trước khi thực hiện truy vấn phân tán

Để bắt đầu cho hai máy có thể truy vấn qua lại, ta cần đảm bảo một số yêu cầu như sau:

- Máy chủ và máy trạm đều phải connect vào Radmin VPN
- Máy chủ đã đăng nhập vào Couchbase bằng tài khoản Couchbase máy chủ
- Máy trạm đã đăng nhập vào tài khoản Couchbase máy chủ tại máy trạm
- Máy chủ đã Rebalance cho các máy trạm đã đăng ký kết nối

Nếu một trong bốn yêu cầu trên không được thoả mãn, Couchbase sẽ từ chối cho các máy trạm truy cập dữ liệu như hình bên dưới



Hình 3. Lỗi “Failure contacting server” khi máy trạm cố gắng truy cập khi không thoả cả 4 điều kiện trên

4.2. Truy vấn dữ liệu phân tán bằng Query Workbench của Couchbase dùng N1QL Query

Liệt kê các sân bay (airports) và địa điểm du lịch (landmarks) trong cùng một thành phố, với các thành phố này thuộc nước Mỹ, sắp xếp theo địa điểm du lịch.

```
SELECT DISTINCT
  MIN(airport.airportname) AS Airport__Name,
  MIN(lmark.name) AS Landmark_Name,
  MIN(airport.tz) AS Landmark_Time
FROM
  `travel-sample`.inventory.airport airport
  RIGHT JOIN `travel-sample`.inventory.landmark lmark
    ON airport.city = lmark.city
    AND airport.country = "United States"
GROUP BY lmark.name
ORDER BY lmark.name;
```

Kết quả câu truy vấn xác định được 4341 documents:

Query Editor < history (4/4) > travel-sample ▼ inventory ▼

```

1 SELECT DISTINCT MIN(airportname) AS Airport__Name,
2                 MIN(lmark.name) AS Landmark_Name,
3                 MIN(airport.tz) AS Landmark_Time
4 FROM `travel-sample`.inventory.airport airport
5 RIGHT JOIN `travel-sample`.inventory.landmark lmark
6 ON airport.city = lmark.city
7 AND airport.country = "United States"
8 GROUP BY lmark.name
9 ORDER BY lmark.name;

```

Execute Explain External Query Advisor ✔ success just now | 12.4s | 4341 docs | 571594 bytes format ↗

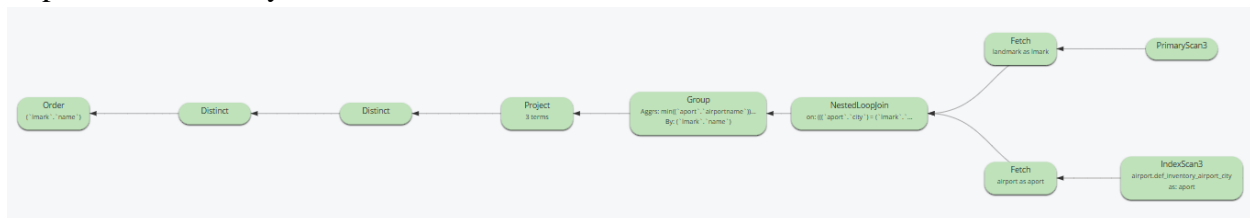
Results Table JSON Chart Plan Plan Text

```

1 {
2   {
3     "Airport__Name": "San Francisco Intl",
4     "Landmark_Name": "&quot;Hippie Temptation&quot; house",
5     "Landmark_Time": "America/Los_Angeles"
6   },
7   {
8     "Airport__Name": null,
9     "Landmark_Name": "'The Argyll Arms Hotel'",
10    "Landmark_Time": null
11  },
12  {
13    "Airport__Name": null,
14    "Landmark_Name": "'Visit the Hut of the Shadows and other End of the Road sculptures",
15    "Landmark_Time": null

```

Explain Plan câu truy vấn:



4.3. Truy vấn dữ liệu phân tán bằng Python dùng NIQL Query

Để thực hiện các thao tác truy vấn dữ liệu phức tạp, ta sẽ ưu tiên sử dụng phương thức truy vấn `couchbase.cluster.Cluster::query()` đã được giới thiệu tại **Yêu cầu 2,3 – phần 2.3**. Và sử dụng câu truy vấn tại **Yêu cầu 2,3 – phần 4.2**.

Trong đó, tại câu lệnh yêu cầu tạo connection, ta thay địa chỉ IP bằng địa chỉ IP của máy trạm, và để tránh quá nhiều kết quả, ta thực hiện giới hạn số lượng kết quả truy vấn là 3.

Ta có đoạn code như sau:

```
import couchbase
from couchbase.cluster import Cluster, ClusterOptions
from couchbase.cluster import QueryOptions
from couchbase.auth import PasswordAuthenticator

cluster = Cluster('couchbase://26.46.70.3',
                  ClusterOptions(
                      PasswordAuthenticator('master', '123456')
                  )
)

result = cluster.query(
    """
    SELECT DISTINCT
        MIN(airportname) AS Airport__Name,
        MIN(lmark.name) AS Landmark_Name,
        MIN(airport.tz) AS Landmark_Time
    FROM
        `travel-sample`.inventory.airport airport
    RIGHT JOIN `travel-sample`.inventory.landmark lmark
        ON airport.city = lmark.city
        AND airport.country = "United States"
    GROUP BY lmark.name
    ORDER BY lmark.name
    LIMIT 3;
    """
)

for row in result:
    print(row)
```

Kết quả câu truy vấn:

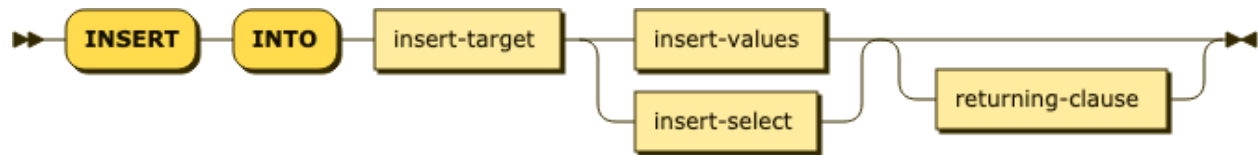
```
[Running] python -u "c:\Users\thaon\Desktop\zehn.py"
{'Airport__Name': 'San Francisco Intl', 'Landmark_Name': '"Hippie Temptation" house', 'Landmark_Time': 'America/Los_Angeles'}
{'Airport__Name': None, 'Landmark_Name': '"The Argyll Arms Hotel"', 'Landmark_Time': None}
{'Airport__Name': None, 'Landmark_Name': '"Visit the Hut of the Shadows and other End of the Road sculptures"', 'Landmark_Time': None}

[Done] exited with code=0 in 13.573 seconds
```

5. Thao tác thêm, xoá, sửa dữ liệu giữa hai máy bằng NoSQL

Ngoài 2 cách được giới thiệu bên dưới, ta có thể chỉnh sửa trực tiếp bằng giao diện của Couchbase Workbench tại máy trạm.

5.1. Thao tác thêm dữ liệu



(Xem chi tiết tại <https://docs.couchbase.com/server/current/n1ql/n1ql-language-reference/insert.html>)

Ví dụ: Thực hiện chạy câu lệnh sau tại Couchbase Workbench của máy trạm.

Câu lệnh sau sẽ thực hiện một dữ liệu thử nghiệm vào collection “airline” của scope “inventory” thuộc bucket “travel-sample”.

```
INSERT INTO `travel-sample`.inventory.airline ( KEY, VALUE )
VALUES ( "airline-test", {
    "id": 9999,
    "type": "airport",
    "airportname": "Test",
    "city": "Test",
}
)
```

- Kết quả chạy câu lệnh:

Query Editor < history (6/6) > travel-sample

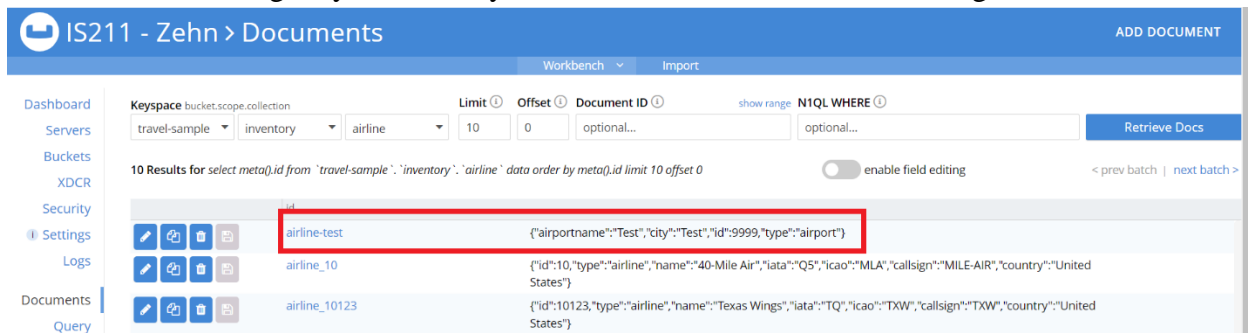
```
1 INSERT INTO `travel-sample`.inventory.airline ( KEY, VALUE )
2 VALUES ( "airline-test", {
3     "id": 9999,
```

Execute **Explain** External Query Advisor ✓ success just now | 2.2s | 1 mutations format

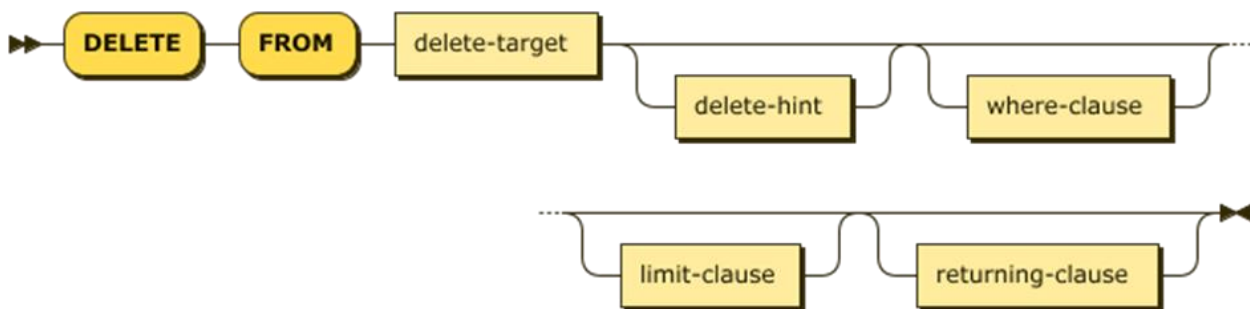
Results Table **JSON** Chart Plan Plan Text

```
1 {
2   "results": []
3 }
```

- Kiểm tra bằng máy chủ, ta thấy dữ liệu đã được thêm vào thành công



5.2. Thao tác xoá dữ liệu



(Xem chi tiết tại [DELETE | Couchbase Docs](#))

5.2.1. Thao tác xoá dữ liệu có điều kiện

Ví dụ: Thực hiện chạy câu lệnh sau tại Couchbase Workbench của máy trạm.

Thực hiện xoá các document hotel khách sạn tại thành phố San Francisco tại collection “hotel” của scope “inventory”.

```
DELETE FROM `travel-sample`.inventory.hotel
WHERE city = "San Francisco";
```

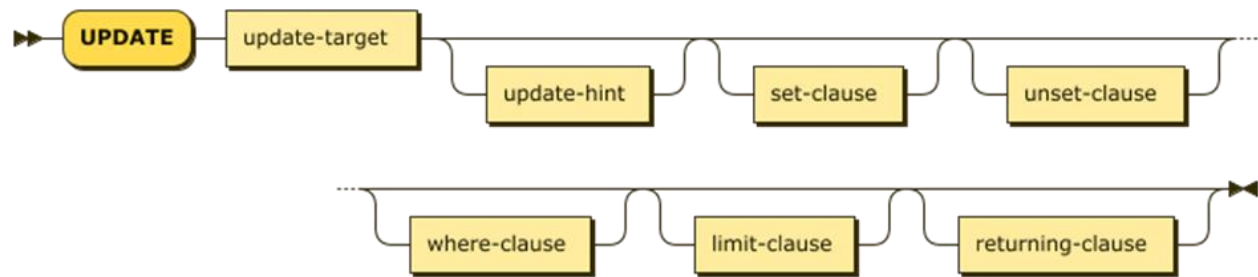
5.2.2. Thao tác xoá dữ liệu sử dụng ID của document

Ví dụ: Thực hiện chạy câu lệnh sau tại Couchbase Workbench của máy trạm.

Thực hiện xoá các document airline có ID document là “airline-test” tại collection “airline” của scope “inventory”.

```
DELETE FROM `travel-sample`.inventory.airline
USE KEYS "airline-test";
```


5.3. Thao tác sửa dữ liệu



(Xem chi tiết tại [UPDATE / Couchbase Docs](#))

5.3.1. Thao tác sửa dữ liệu dùng ID document

Ví dụ: Thực hiện chạy câu lệnh sau tại Couchbase Workbench của máy trạm.

Sửa nickname của landmark có ID là 10090 thành “Squiggly Bridge” tại collection “landmark” của scope “inventory”.

```
UPDATE `travel-sample`.inventory.landmark
  USE KEYS "landmark_10090"
  SET nickname = "Squiggly Bridge";
```

5.3.2. Thao tác sửa dữ liệu dùng Sub-query

Ví dụ: Thực hiện chạy câu lệnh sau tại Couchbase Workbench của máy trạm.

Thêm thông tin khách sạn cho sân bay có “faa” là “NCE” và thông tin những khách sạn sẽ thêm vào sẽ là tên khách sạn và ID của khách sạn đó với những khách sạn ở thành phố “Nice”.

```
UPDATE `travel-sample`.inventory.airport AS a
SET hotels = (
  SELECT h.name, h.id
  FROM `travel-sample`.inventory.hotel AS h
  WHERE h.city = "Nice"
)
WHERE a.faa = "NCE"
```

Phụ lục:

Tham khảo

[1] Couchbase, NoSQL Companies CouchOne and Membase Merge to Form Couchbase

<https://www.couchbase.com/press-releases/membase-couchone-merge>

[2] Couchbase Documentation, *Travel App Data Model*

<https://docs.couchbase.com/java-sdk/current/ref/travel-app-data-model.html>

[3] About Couchbase

<https://www.couchbase.com/about>

Các đường dẫn liên quan

Github: https://github.com/tori4582/phanta_IS221.M11_10