

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN



BÀI TẬP LỚN MÔN HỌC
CƠ SỞ DỮ LIỆU PHÂN TÁN

ĐỀ TÀI:

TÌM HIỂU HỆ QUẢN TRỊ CSDL NOSQL – RETHINKDB

Giảng viên hướng dẫn: **Nguyễn Minh Nhật**

Lớp: **IS211.M11.HTCL**

Nhóm sinh viên thực hiện: **Peekaboo Team**

STT	Họ và Tên	MSSV
1	Tạ Quang Huy	19520113
2	Lê Thị Hồng Cúc	19521295
3	Chu Xuân Sơn	19522130
4	Nguyễn Ảnh Trường Thắng	19522204

TP. Hồ Chí Minh, tháng 9 năm 2021

LỜI CẢM ƠN

Lời đầu tiên, nhóm em chân thành cảm ơn anh Nguyễn Minh Nhật đã cung cấp cho chúng em những kiến thức để chúng em có thể thực hiện được bài tập này, cũng như sự góp ý giúp đỡ rất nhiệt tình, chân thành và luôn giải đáp kịp thời những thắc mắc của chúng em. Nếu không có những lời hướng dẫn của anh Nhật thì em nghĩ bài báo cáo của nhóm rất khó để hoàn thành được.

Đây cũng là cơ hội để từng thành viên của nhóm được làm việc với những bạn mới, học thêm các kĩ năng làm việc nhóm, học hỏi lẫn nhau và quan trọng là có cơ hội được thực hiện sản phẩm thông qua môn học.

Trong thời gian thực hiện đồ án, nhóm đã vận dụng những kiến thức đã được dạy, đồng thời sử dụng những cái mới, với mong muốn có thể hoàn thiện bài làm một cách chu đáo nhất. Tuy nhiên với thời gian, kiến thức và kinh nghiệm còn hạn chế nên không thể tránh khỏi những thiếu sót, vì vậy nhóm em rất mong nhận được những góp ý quý báu từ phía các Thầy cô, anh chị đi trước, giúp nhóm có điều kiện bổ sung, nâng cao kiến thức phục vụ tốt hơn cho những đồ án cũng như công tác thực tế sau này.

Sau cùng, nhóm em kính chúc anh có thật nhiều sức khỏe trong mùa dịch COVID – 19 phức tạp như hiện nay, để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Nhóm thực hiện,

Quang Huy – Hồng Cúc – Xuân Sơn – Trường Thắng.

MỤC LỤC

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	3
1. Giới thiệu về NoSQL	3
2. Giới thiệu cơ bản về hệ quản trị CSDL RethinkDB.....	6
2.1. Lịch sử ra đời nguồn gốc.....	6
2.2. Tổng quan về hệ quản trị CSDL RethinkDB	7
2.3. Khi nào nên dùng RethinkDB	9
2.4. Các giao tác trong RethinkDB.....	9
CHƯƠNG 2: CÀI ĐẶT TRÊN 2 MÁY TRỞ LÊN VÀ THỰC HIỆN TRUY VẤN GIỮA HAI MÁY.....	31
1. Cài đặt RethinkDB.....	31
2. Cơ chế phân tán của RethinkDB.....	34
3. Kết nối nhiều máy thông qua mạng LAN ảo	38
CHƯƠNG 3: THAO TÁC DỮ LIỆU QUA NHIỀU MÁY.....	41
1. Thao tác với RethinkDB	41
2. Kết quả.....	58
3. Cơ chế nhân bản trong phân tán RethinkDB.....	72
TÀI LIỆU THAM KHẢO.....	75

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1. Giới thiệu về NoSQL

Cơ sở dữ liệu NoSQL là Cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn.

Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.

Cơ sở dữ liệu NoSQL là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ liệu cực kỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao để đem đến cho người dùng trải nghiệm tuyệt vời.

- **Linh hoạt:** Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.
- **Khả năng thay đổi quy mô:** Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém. Một số

nhà cung cấp dịch vụ đám mây xử lý các hoạt động này một cách không công khai dưới dạng dịch vụ được quản lý đầy đủ.

- Hiệu năng cao: Cơ sở dữ liệu NoSQL được tối ưu hóa theo các mô hình dữ liệu cụ thể và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.
- Cực kỳ thiết thực: Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

Tại sao lại cần phải có NoSQL.

Sở dĩ người ta phát triển NoSQL xuất phát từ yêu cầu cần những database có khả năng lưu trữ dữ liệu với lượng cực lớn, truy vấn dữ liệu với tốc độ cao mà không đòi hỏi quá nhiều về năng lực phần cứng cũng như tài nguyên hệ thống và tăng khả năng chịu lỗi.

Đây là những vấn đề mà các relational database không thể giải quyết được.

Lượng dữ liệu mà các hệ thống cần phải xử lý giờ đây ngày 1 lớn. Ví dụ như Google, Facebook phải lưu trữ và xử lý một lượng dữ liệu cực lớn mỗi ngày .

Ưu và Nhược điểm của NoSQL

Ưu điểm

Có một số lợi thế khi làm việc với cơ sở dữ liệu NoSQL như MongoDB và Cassandra. Những ưu điểm chính là khả năng mở rộng và tính sẵn sàng cao.

High scalability: Cơ sở dữ liệu NoSQL sử dụng sharding để chia tỷ lệ ngang. Việc phân vùng dữ liệu và đặt nó trên nhiều máy sao cho thứ tự của dữ liệu được bảo toàn là sắc nét hơn. Chia tỷ lệ dọc có nghĩa là thêm nhiều tài nguyên hơn vào máy hiện có trong khi chia tỷ lệ ngang có nghĩa là thêm nhiều máy hơn để xử lý dữ liệu. Chia tỷ lệ dọc không dễ thực hiện nhưng chia tỷ lệ ngang rất dễ thực hiện. Ví dụ về cơ sở dữ liệu mở rộng theo chiều ngang là MongoDB, Cassandra, v.v. NoSQL có thể xử lý lượng dữ liệu khổng lồ do khả năng mở rộng, khi dữ liệu tự phát triển NoSQL tự mở rộng quy mô để xử lý dữ liệu đó một cách hiệu quả.

High availability: Tính năng tự động sao chép trong MongoDB làm cho nó rất sẵn sàng vì trong trường hợp có bất kỳ lỗi nào, dữ liệu sẽ tự sao chép về trạng thái nhất quán trước đó.

Nhược điểm

Bên cạnh những ưu điểm thì NoSQL Database cũng có những nhược điểm sau:

Quản lý dữ liệu: Mục đích của các công cụ dữ liệu lớn là làm cho việc quản lý một lượng lớn dữ liệu trở nên đơn giản nhất có thể. Nhưng nó không phải là dễ dàng như vậy. Quản lý dữ liệu trong NoSQL phức tạp hơn nhiều so với cơ sở dữ liệu quan hệ. Đặc biệt, NoSQL nổi tiếng là khó cài đặt và thậm chí là để quản lý hằng ngày cũng tốn khá nhiều thời gian.

Sao lưu dữ liệu: Sao lưu là một điểm yếu lớn đối với một số cơ sở dữ liệu NoSQL như MongoDB. MongoDB không có cách tiếp cận để sao lưu dữ liệu một cách nhất quán.

Thiếu tính nhất quán: NoSQL đánh đổi sự nhất quán để ưu tiên tốc độ, hiệu suất hiệu quả hơn.

Trọng tâm hẹp: Cơ sở dữ liệu NoSQL có trọng tâm rất hẹp vì nó chủ yếu được thiết kế để lưu trữ nhưng nó cung cấp rất ít chức năng. Cơ sở dữ liệu quan hệ là lựa chọn tốt hơn trong lĩnh vực Quản lý giao dịch so với NoSQL.

Mã nguồn mở: NoSQL là cơ sở dữ liệu mã nguồn mở và không có tiêu chuẩn đáng tin cậy cho NoSQL được nêu ra.

Không có lược đồ: Ngay cả khi bạn lấy dữ liệu ở dạng tự do, bạn hầu như luôn cần áp đặt các ràng buộc để làm cho nó hữu ích. Với NoSQL, trách nhiệm sẽ được chuyển từ cơ sở dữ liệu sang nhà phát triển ứng dụng. Ví dụ, nhà phát triển có thể áp đặt cấu trúc thông qua một hệ thống map đối tượng quan hệ hoặc ORM. Nhưng nếu bạn muốn lược đồ tự dữ liệu, NoSQL thường sẽ không hỗ trợ.

2. Giới thiệu cơ bản về hệ quản trị CSDL RethinkDB

2.1. Lịch sử ra đời nguồn gốc

RethinkDB được thành lập vào năm 2009 và là mã nguồn mở ở phiên bản 1.2 vào năm 2012. Ban đầu, RethinkDB được tài trợ bởi Y Combinator vào tháng 6 năm 2009.

Vào năm 2015, RethinkDB đã phát hành phiên bản 2.0 và sẵn sàng được sản xuất.

Vào ngày 5 tháng 10 năm 2016, công ty thông báo ngừng hoạt động, chuyển các thành viên trong nhóm kỹ sư của mình sang Stripe và sẽ không cung cấp hỗ trợ sản xuất nữa. Công ty đã thông báo rằng họ không thể xây dựng một doanh

nghiệp bền vững và các sản phẩm của họ trong tương lai sẽ hoàn toàn là nguồn mở mà không có hỗ trợ thương mại.

Vào ngày 6 tháng 2 năm 2017, Cloud Native Computing Foundation đã mua quyền đối với mã nguồn và cấp phép mã nguồn theo Giấy phép Apache 2.0.

2.2. Tổng quan về hệ quản trị CSDL RethinkDB

RethinkDB là một cơ sở dữ liệu phân tán NoSQL – hướng tài liệu (document database). RethinkDB là cơ sở dữ liệu mã nguồn mở miễn phí được tạo ra bởi một công ty cùng tên.



RethinkDB lưu trữ các tài liệu JSON với các lược đồ động và được thiết kế để tạo điều kiện thuận lợi cho việc đẩy các bản cập nhật theo thời gian thực cho các kết quả truy vấn tới các ứng dụng. Dữ liệu được thêm vào lưu trữ dưới dạng cấu trúc JSON tự do hoặc “tài liệu”, trong đó dữ liệu có thể là bất kỳ kiểu nào, từ số nguyên đến chuỗi hay đến các văn bản tự do.

Hệ quản trị CSDL RethinkDB có khả năng mở rộng được phạm vi, được thiết kế cho nền tảng của các ứng dụng thời gian thực. RethinkDB cho phép lập trình viên có thể xây dựng được các ứng dụng thời gian thực có thể mở rộng phạm vi trong một khoảng thời gian với công sức là ít nhất.

RethinkDB sử dụng ngôn ngữ truy vấn ReQL, một ngôn ngữ cụ thể được sử dụng riêng cho Ruby, Python, Java và JavaScript (bao gồm Node JS). Nó

có hỗ trợ cho các phép kết bảng (join), nhóm (group by), tổng hợp (aggregations). Ngoài ra còn có các trình điều khiển không chính thức, được cộng đồng hỗ trợ cho các ngôn ngữ khác, bao gồm C #, Clojure, Erlang, Go, Haskell, Lua và PHP.

Ưu điểm và nhược điểm

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> + Là một cơ sở dữ liệu mã nguồn mở cho các ứng dụng web. + Sử dụng một tài liệu động JSON là NoSQL. + Sử dụng cơ sở dữ liệu phân tán nên rất dễ mở rộng. + Cung cấp một cơ sở dữ liệu có tính khả dụng cao với các chức năng tự động. 	<ul style="list-style-type: none"> + Trong RethinkDB, mỗi bảng yêu cầu không gian đĩa tối thiểu là 10 MB. + Các khóa chính bị giới hạn trong RethinkDB.

So sánh các thuật ngữ SQL với RethinkDB:

SQL	RethinkDB
database	database
table	table
row	document
column	field
table joins	table joins
primary key	primary key (by default id)

2.3. Khi nào nên dùng RethinkDB

Mô hình truy cập cơ sở dữ liệu request - response hoạt động tốt trên web bởi vì nó ánh xạ trực tiếp đến giao thức HTTP request – response. Các ứng dụng hiện đại yêu cầu gửi dữ liệu trực tiếp cho khách hàng trong thời gian thực (real-time), lập trình viên có thể yêu cầu RethinkDB liên tục đẩy các kết quả truy vấn cập nhật đến các ứng dụng trong thời gian thực

RethinkDB được dùng cho các ứng dụng yêu cầu những mối quan hệ chặt chẽ từ nguồn cấp dữ liệu thời gian thực.

Cụ thể, RethinkDB có thể được dùng để xây dựng

- Ứng dụng dành cho thiết bị di động và web cộng tác.
- Ứng dụng truyền tải và phân tích trực tuyến.
- Trò chơi nhiều người chơi.
- Thương mại điện tử thời gian thực.
- Các thiết bị đã được kết nối.

2.4. Các giao tác trong RethinkDB

2.4.1. Tạo cơ sở dữ liệu

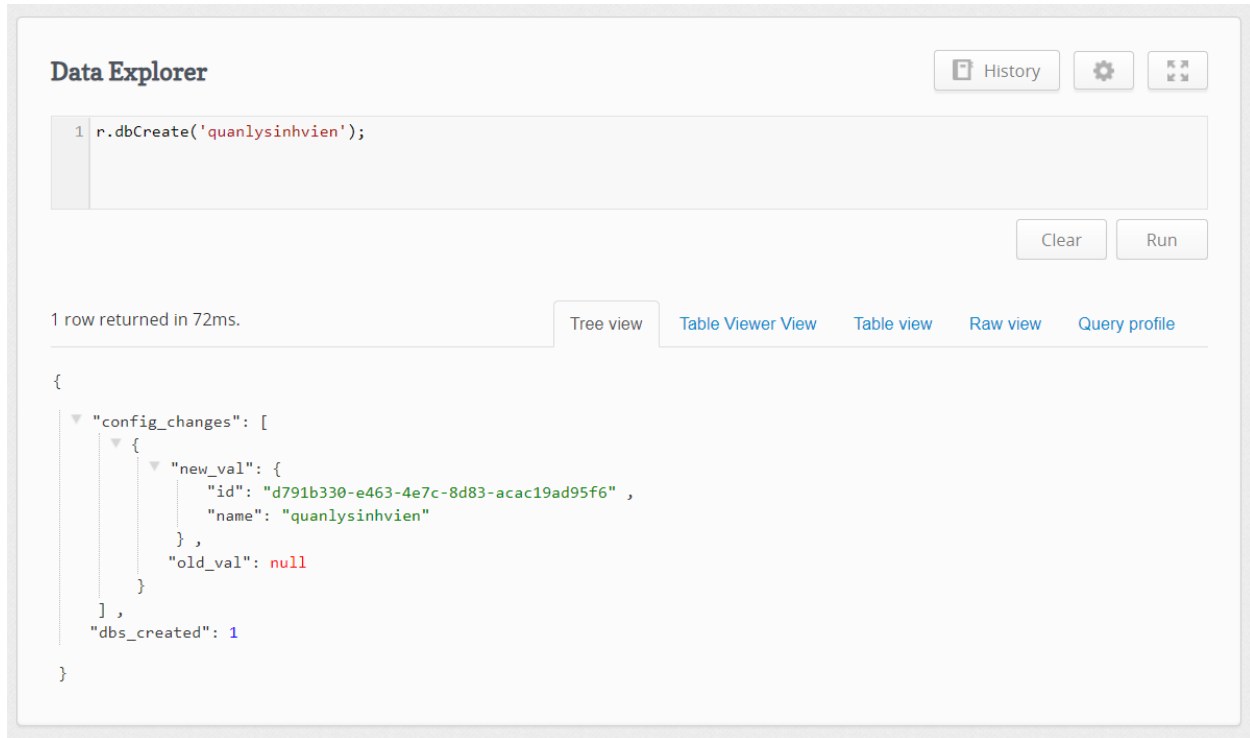
Cách 1: Sử dụng Data Explorer (Sử dụng ngôn ngữ ReQL)

Cú pháp: `r.dbCreate('<TEN DATABASE>');`

So sánh giữa SQL và ReQL

SQL	ReQL
<code>CREATE DATABASE my_database;</code>	<code>r.dbCreate('my_database')</code>

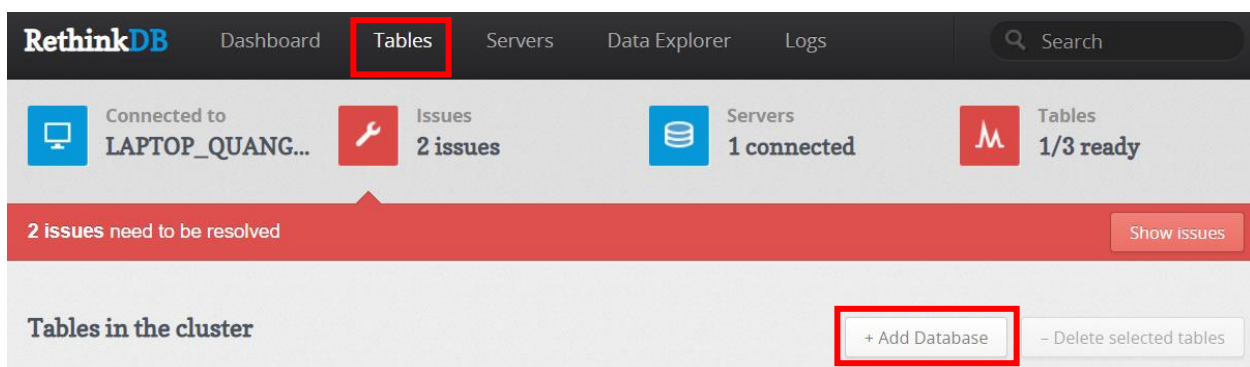
Ví dụ:



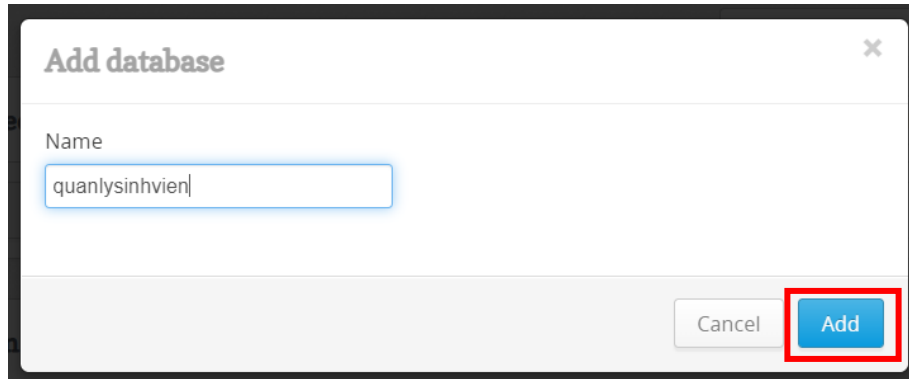
Cách 2: Sử dụng giao diện RethinkDB.

Bước 1: Chọn "Tables" trên thanh menu.

Bước 2: Chọn "+Add Database"



Bước 3: Đặt tên cho database và chọn “Add” để tạo CSDL.



2.4.2. Xóa cơ sở dữ liệu

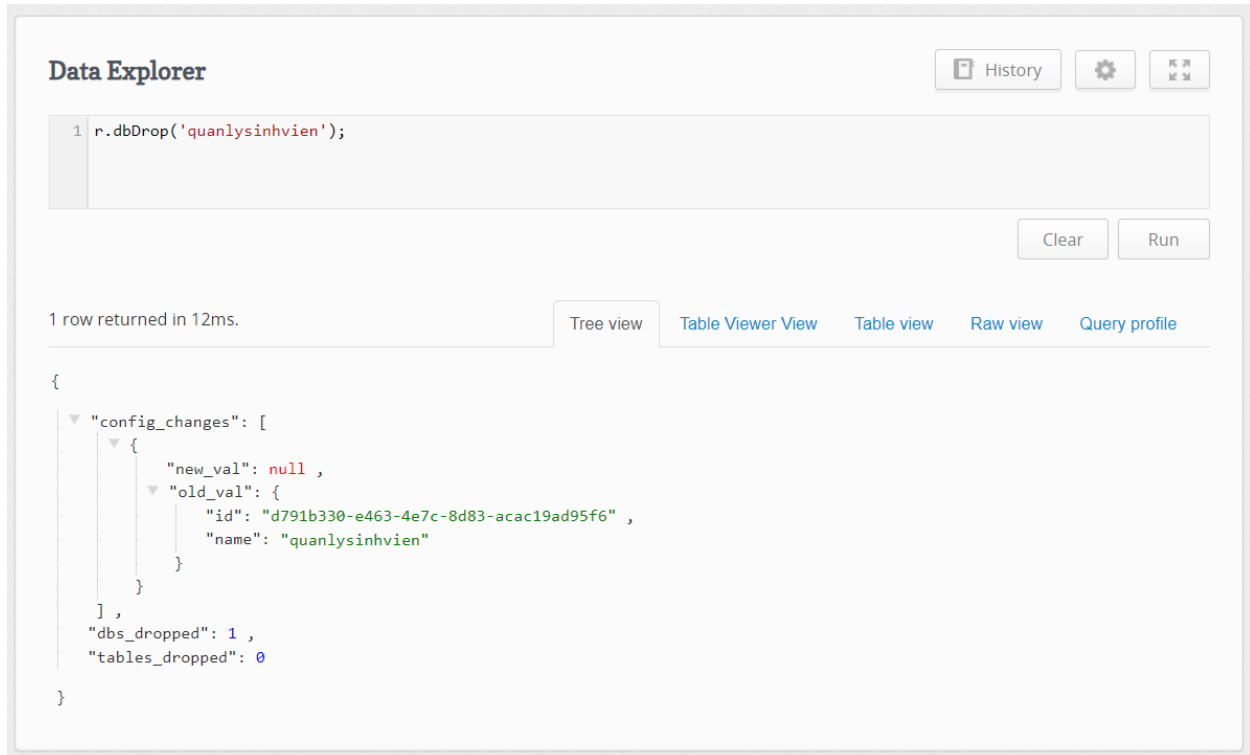
Cách 1: Sử dụng Data Explorer (Sử dụng ngôn ngữ ReQL)

Cú pháp: `r.dbDrop('<TEN DATABASE>');`

So sánh giữa SQL và ReQL

SQL	ReQL
<code>DROP DATABASE my_database;</code>	<code>r.dbDrop('my_database')</code>

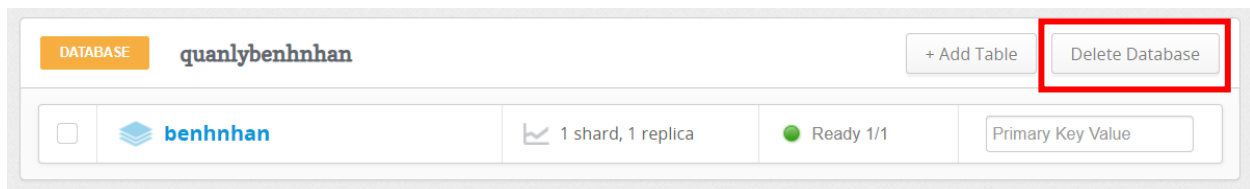
Ví dụ:



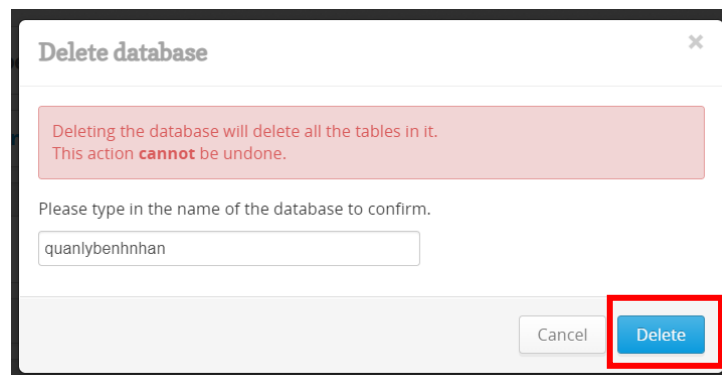
Cách 2: Sử dụng giao diện RethinkDB.

Bước 1: Chọn "Tables" trên thanh menu.

Bước 2: Chọn "Delete Database"



Bước 3: Nhập lại tên database để xác nhận và chọn "Delete" để xóa database.



2.4.3. Tạo bảng

Cách 1: Sử dụng Data Explorer (Sử dụng ngôn ngữ ReQL)

Cú pháp:

```
r.tableCreate('<TEN DATABASE>');
```

```
r.tableCreate('<TEN DATABASE>', {primaryKey: "<TEN KHOA CHINH>"});
```

So sánh giữa SQL và ReQL

SQL	ReQL
<pre>CREATE TABLE SinhVien (MaSV INT IDENTITY(1,1) PRIMARY KEY, TenSV VARCHAR(50), GioiTinh INT);</pre>	<pre>r.db('my_db').tableCreate('SinhVien', {primaryKey: "MaSV"})</pre> <p>Lưu ý:</p> <ul style="list-style-type: none">- RethinkDB là một cơ sở dữ liệu NoSQL và không có lược đồ (schema).- Nếu không cài đặt khóa chính thì mặc định khóa chính là <code>id</code>.

Ví dụ:

Data Explorer History Settings Fullscreen

```
1 r.db('quanlysinhvien').tableCreate('sinhvien', {primaryKey: "MaSV"});
```

Clear Run

1 row returned in 1.16s. Tree view Table Viewer View Table view Raw view Query profile

```
{
  "config_changes": [
    {
      "new_val": {
        "db": "quanlysinhvien",
        "durability": "hard",
        "id": "0885f402-15ba-4151-a919-678cc80ec589",
        "indexes": [ ],
        "name": "sinhvien",
        "primary_key": "MaSV",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "LAPTOP_QUANGHUY",
            "replicas": [
              "LAPTOP_QUANGHUY"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      },
      "old_val": null
    }
  ],
  "tables_created": 1
}
```

Cách 2: Sử dụng giao diện RethinkDB.

Bước 1: Chọn "Tables" trên thanh menu.

Bước 2: Tại database, chọn "+Add Table"

DATABASE **quanlysinhvien** + Add Table Delete Database

There are no tables in this database.

Bước 3: Nhập tên bảng, nhập khóa chính (optional) và chọn “Create table” để tạo bảng.

Add table to quanlysinhvien

Name

sinhvien

Hide optional settings

Primary key

MaSV

Acknowledge writes only when written to disk

YES

NO

Cancel

Create table

2.4.4. Xóa bảng

Cách 1: Sử dụng Data Explorer (Sử dụng ngôn ngữ ReQL)

Cú pháp: `r.tableDrop('<TEN DATABASE>');`

So sánh giữa SQL và ReQL

SQL	ReQL
<code>DROP TABLE KhuCachLy;</code>	<code>r.db('my_db').tableDrop("KhuCachLy")</code>

Ví dụ:

Data Explorer History Settings Fullscreen

```
1 r.db('quanlysinhvien').tableDrop("sinhvien");
```

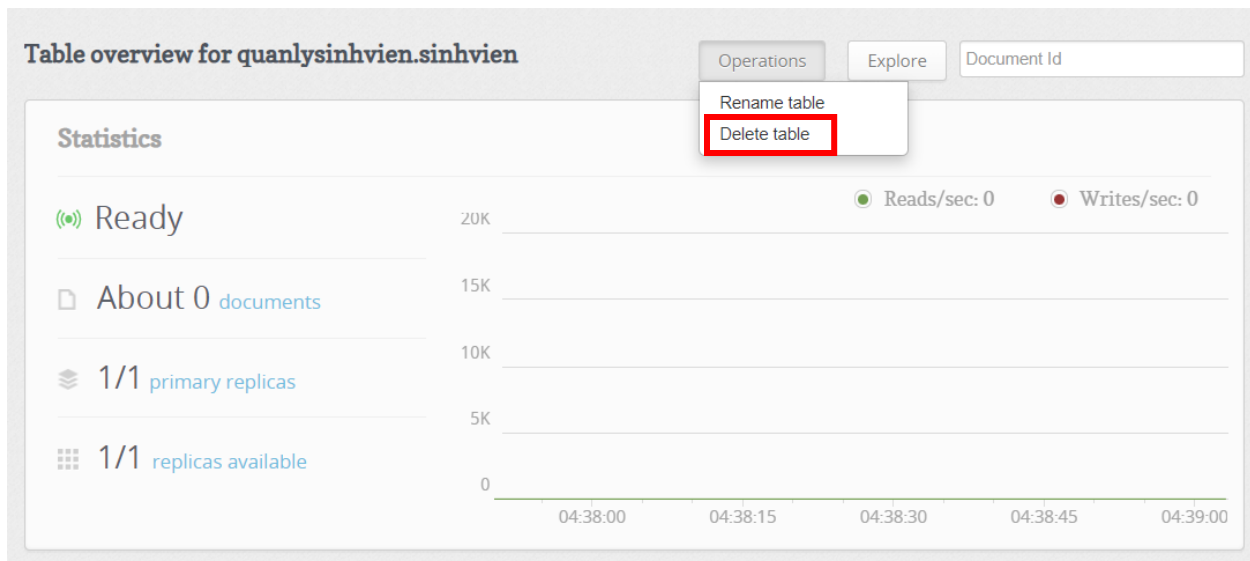
Clear Run

1 row returned in 796ms. Tree view Table Viewer View Table view Raw view Query profile

```
{
  "config_changes": [
    {
      "new_val": null,
      "old_val": {
        "db": "quanlysinhvien",
        "durability": "hard",
        "id": "0885f402-15ba-4151-a919-678cc80ec589",
        "indexes": [ ],
        "name": "sinhvien",
        "primary_key": "MaSV",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "LAPTOP_QUANGHUY",
            "replicas": [
              "LAPTOP_QUANGHUY"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      }
    ]
  ],
  "tables_dropped": 1
}
```

Cách 2: Sử dụng giao diện RethinkDB.

Bước 1: Chọn bảng cần xóa



Bước 2: Chọn Operation > Delete table > Delete

Delete tables ×

Deleting a table will delete all its data. This action cannot be reversed.

Are you sure you want to delete this table:

- quanlysinhvien.sinhvien

Cancel Delete

2.4.5. Thêm dữ liệu vào bảng

Cú pháp: `r.table('<TEN DATABASE>').insert(document);`

So sánh giữa SQL và ReQL:

SQL	ReQL
<pre>INSERT INTO SanPham(MaSP, TenSP, SL) VALUES ("SP01", "Bút chì", 30)</pre>	<pre>r.table("SanPham").insert({ "MaSP": "SP01", "TenSP": "Bút chì", "SL": 30 })</pre>

Các ví dụ cụ thể:

The screenshot shows the Data Explorer interface. At the top, there's a title bar with 'Data Explorer' and buttons for 'History', 'Settings', and 'Fullscreen'. Below the title bar is a code editor with a ReQL query:

```
1 r.db('quanlysinhvien').table('sinhvien').insert({
2   "MaSV": "SV01",
3   "HoTen": "Lê Thị Hồng Cúc",
4   "Lop": "HTCL2019",
5   "DTB": 8.8
6 }
7 });
```

 To the right of the code editor are 'Clear' and 'Run' buttons. Below the code editor, it says '1 row returned in 101ms.' and there are tabs for 'Tree view', 'Table Viewer View', 'Table view', 'Raw view', and 'Query profile'. The 'Table Viewer View' tab is selected, showing a JSON response:

```
{
  "deleted": 0,
  "errors": 0,
  "inserted": 1,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}
```

Thêm nhiều document cùng một lúc

Data Explorer

History

```

1 r.db('quanlysinhvien').table('sinhvien').insert([
2   {
3     "MaSV": "SV02",
4     "HoTen": "Chu Xuân Sơn",
5     "Lop": "HTCL2019",
6     "DTB": 8.8,
7     "DiemThi": { "CSDL phân tán": "10", "Nhập môn lập trình": "8", "ERP": "9" },
8   },
9   {
10    "MaSV": "SV03",
11    "HoTen": "Nguyễn Ảnh Trường Thắng",
12    "Lop": "HTCL2019",
13    "CacMonDaHoc": ["Lập trình web", "Dữ liệu lớn", "Mạng xã hội"],
14  })

```

Clear Run

1 row returned in 66ms.

Tree view Table Viewer View Table view Raw view Query profile

```

{
  "deleted": 0,
  "errors": 0,
  "inserted": 2,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}

```

2.4.6. Truy vấn dữ liệu

So sánh giữa SQL và ReQL và các ví dụ:

SQL	ReQL
SELECT * FROM SinhVien	<code>r.table("SinhVien")</code>
SELECT MaSV, TenSV FROM SinhVien	<code>r.table("SinhVien").pluck("MaSV", "TenSV")</code>
SELECT * FROM SinhVien WHERE TenSV = "Võ Ngọc Cẩm Tú"	Cách 1: <code>r.table("SinhVien").filter({ "TenSV": "Võ Ngọc Cẩm Tú" })</code> Cách 2:

	<pre>r.table("SinhVien") .get_all("Võ Ngọc Cẩm Tú", index="TenSV")</pre>
<pre>SELECT * FROM BenhNhan WHERE TenBN = "Nguyễn Văn Hiếu" AND Tuoi = 20</pre>	<pre>r.table("BenhNhan").filter({ "TenBN": "Nguyễn Văn Hiếu", "Tuoi": 20 })</pre>
<pre>SELECT * FROM PhongHoc WHERE MaPhong LIKE "E1%"</pre>	<pre>r.table("PhongHoc").filter(r.row['MaPhong'].match("^E1"))</pre>
<pre>SELECT * FROM HoaDon ORDER BY TriGia ASC</pre>	<pre>r.table("HoaDon").order_by("TriGia")</pre>
<pre>SELECT MaNV FROM NhanVien WHERE MaPB = "P.KeToan" ORDER BY Luong DESC</pre>	<pre>r.table("NhanVien").filter({ "MaPB": "P.KeToan" }).order_by(r.desc("Luong"))</pre>
<pre>SELECT * FROM SanPham LIMIT 5 SKIP 10</pre>	<pre>r.table("SanPham").skip(10).limit(5)</pre>
<pre>SELECT * FROM KhamBenh WHERE ChanDoan IN ('Sốt xuất huyết', 'Đau răng')</pre>	<pre>r.table("KhamBenh").filter(lambda doc: r.expr(["Sốt xuất huyết", "Đau răng"]) .contains(doc["ChanDoan"]))</pre>

```
SELECT * FROM KhachHang
WHERE TenKH NOT IN ('Trần Đức Bo',
'Nguyễn Kim Tuyến')
```

```
r.table("KhachHnag").filter(lambda doc:
    r.expr(["Trần Đức Bo", "Nguyễn Kim
Tuyến"])
    .contains(doc["TenKH"])
    .not_()
)
```

```
SELECT COUNT(*) FROM users
```

```
r.table("users").count()
```

```
SELECT COUNT(MaBN) FROM BenhNhan
WHERE Tuổi > 60
```

```
r.table("BenhNhan").filter(
    (r.row.has_fields("MaBN"))
    & (r.row["Tuoi"] > 60)
).count()
```

```
SELECT AVG("Tuoi")
FROM BenhNhan
```

```
r.table("BenhNhan")
    .avg("Tuoi")
```

```
SELECT MAX("DiemTB")
FROM SinhVien
```

```
r.table("SinhVien")["DiemTB"]
    .max()
```

```
SELECT DISTINCT(name) FROM users
```

```
r.table("users").pluck("name")
    .distinct()
```

```
SELECT *
FROM LopHoc
WHERE SiSo BETWEEN 20 AND 60;
```

```
r.table("LopHoc").filter(
    (r.row["SiSo"] >= 20)
    & (r.row["SiSo"] <= 60))
```

```
SELECT name, 'is_adult' = CASE
  WHEN age > 18 THEN 'yes'
  ELSE 'no'
END
FROM users
```

```
r.table("users").map({
  "name": r.row["name"],
  "is_adult": r.branch(
    r.row["age"] > 18,
    "yes",
    "no"
  )
})
```

```
SELECT *
FROM posts
WHERE EXISTS
  (SELECT * FROM users
   WHERE posts.author_id
     = users.id)
```

```
r.table("posts")
  .filter(lambda post:
    r.table("users")
      .filter(lambda user:
        user.id == post.author_id
      ).count() > 0
  )
```

Ngoài ra, để truy vấn dữ liệu, ta còn có thể sử dụng hàm `get()`

The screenshot shows the 'Data Explorer' interface. At the top, there are buttons for 'History', a settings gear, and a refresh icon. Below these is a text area containing the query: `1 r.db('quanlysinhvien').table('sinhvien').get("SV02")`. To the right of the text area are 'Clear' and 'Run' buttons. Below the text area, it says '1 row returned in 29ms.' and there are tabs for 'Tree view', 'Table Viewer View', 'Table view', 'Raw view', and 'Query profile'. The 'Tree view' tab is selected, showing a JSON document structure:

```
{
  "DTB": 8.8 ,
  "DiemThi": {
    "CSDL phân tán": "10" ,
    "ERP": "9" ,
    "Nhập môn lập trình": "8"
  } ,
  "HoTen": "Chu Xuân Sơn" ,
  "Lop": "HTCL2019" ,
  "MaSV": "SV02"
}
```

Nếu chúng ta muốn lấy `n` document, chúng ta có thể sử dụng hàm `limit()`.

Data Explorer

History

```
1 r.db('quanlybenhnan').table('benhnan').limit(2)
```

ClearRun

2 rows returned. Displaying rows 0-1

Tree view

Table Viewer View

Table view

Raw view

Query profile

Sử dụng `filter()`, chúng ta có thể nhận được tất cả các tài liệu thỏa mãn một điều kiện.

Data Explorer

History

```
1 r.db('quanlybenhnan').table('benhnan').filter({"Tuoi": 20})
```

ClearRun

1 row returned.

Tree view

Table Viewer View

Table view

Raw view

Query profile

```
{
  "HoTen": "Võ Ngọc Cẩm Tú" ,
  "LichSuKhamBenh": {
    "10-12-2021": {
      "ChanDoan": [
        "Tăng huyết áp" ,
        "Viêm xoang"
      ] ,
      "TenBS": "Nguyễn Xuân Minh Thu"
    } ,
    "20-11-2021": {
      "ChanDoan": [
        "Sốt siêu vi" ,
        "Sâu răng"
      ] ,
      "TenBS": "Trần Thị Ngọc An"
    }
  } ,
  "MaSV": "BN01" ,
  "Tuoi": 20 ,
  "id": "50f2d525-7660-4ec8-9b8e-c301fbe72d2c"
}
```


Để lọc dựa trên hai hoặc nhiều điều kiện:

Data Explorer

History

```
1 r.db('quanlybenhnan').table('benhnan').filter(r.row("HoTen").eq("Nguyễn Hoàng Thắng").or(r.row("Tuoi").eq(24)))
```

ClearRun

1 row returned.

Tree viewTable Viewer ViewTable viewRaw viewQuery profile

```
{
  "HoTen": "Nguyễn Hoàng Thắng",
  "LichSuKhamBenh": {
    "30-11-2021": {
      "ChanDoan": [
        "Viêm ruột thừa"
      ],
      "TenBS": "Trần Thị Ngọc An"
    }
  },
  "MaSV": "BN05",
  "Tuoi": 24,
  "id": "4ce47f69-bc07-4042-9cf7-1ac7579131ca"
}
```

2.4.7. Cập nhật dữ liệu

So sánh giữa SQL và ReQL và các ví dụ:

SQL	ReQL
<pre>UPDATE usersUPDATE SanPham SET TinhTrang = "Hết hàng" WHERE TonKho < 1</pre>	<pre>r.table("users").filter(r.table("San Pham").filter(r.row["TonKho"] < 1)).update({ "TinhTrang": "Hết hàng"</pre>
<pre>UPDATE users SET age = age+1</pre>	<pre>r.table("users").update({ "age": r.row["age"]+1 })</pre>

2.4.6. Xóa dữ liệu

So sánh giữa SQL và ReQL và các ví dụ:

SQL	ReQL
DELETE FROM SinhVien	<code>r.table("SinhVien").delete()</code>
DELETE FROM SanPham WHERE Gia < 10000	<code>r.table("SanPham") .filter(r.row["Gia"] < 10000) .delete()</code>

2.4.6. Kết bảng

So sánh giữa SQL và ReQL và các ví dụ:

SQL	ReQL
SELECT * FROM SinhVien SV JOIN Khoa K ON SV.MaKhoa = K.MaKhoa	<code>r.table("SinhVien").inner_join(r.table("Khoa"), lambda SinhVien, Khoa: SinhVien["MaSV"] == Khoa["MaKhoa"])</code> .zip()
	<p>Sử dụng hàm <code>zip()</code> để kết bảng trong ReQL. Tuy nhiên, nếu cả khóa chính hoặc khóa ngoại được xây dựng trên trường có cùng tên, ta có thể thực hiện phép nối hiệu quả hơn với <code>eq_join</code>.</p>
	<code>r.table("SinhVien").eq_join("ID", r.table("Khoa"), index="ID")</code> .zip()

```

SELECT posts.id AS post_id,
       user.name,
       users.id AS user_id
FROM posts
JOIN users
  ON posts.user_id = users.id

SELECT posts.id AS post_id,
       user.name,
       users.id AS user_id
FROM posts
INNER JOIN users
  ON posts.user_id = users.id

```

```

r.table("posts").inner_join(
  r.table("users"),
  lambda post, user:
    post["user_id"] == user["id"]
).map({
  "post_id": r.row["left"]["id"],
  "user_id": r.row["right"]["id"],
  "name": r.row["right"]["name"]
})

```

```

SELECT *
FROM posts
RIGHT JOIN users
  ON posts.user_id = users.id

SELECT *
FROM posts
RIGHT OUTER JOIN users
  ON posts.user_id = users.id

```

```

r.table("posts").outer_join(
  r.table("users"),
  lambda post, user:
    post["user_id"] == user["id"]
).zip()

```

```

SELECT *
FROM posts
LEFT JOIN users
  ON posts.user_id = users.id

```

```

r.table("users").outer_join(
  r.table("posts"),
  lambda user, post:
    post.user_id == user.id
).zip()

```

```
SELECT *
FROM posts
LEFT OUTER JOIN users
ON posts.user_id = users.id
```

```
r.table("users").concat_map(lambda
user:
    r.table("posts")
    .get_all(user["id"],index="id")
    .do( lambda results:
        r.branch(
            results.count() == 0,
            [{"left": user}],
            results.map( lambda post:
                {
                    "left": user
                    "right": post
                }
            )
        )
    )
).zip()
```

Ví dụ: Kết quả Join bảng Khoa và sinhvien

Data Explorer

History

```

1 r.db("quanlysinhvien").table("Khoa").innerJoin(
2   r.db("quanlysinhvien").table("sinhvien").filter({
3     "HoTen": "Lê Thị Hồng Cúc"
4   }),function(sinhvien, Khoa){
5     return sinhvien("MaKhoa").eq(Khoa("MaKhoa"))
6   }).zip()

```

Clear

Run

1 row returned.

Tree view

Table Viewer View

Table view

Raw view

Query profile

```

{
  "DTB": 8.8 ,
  "GiangVien": [
    "Nguyễn Đình Thuận" ,
    "Nguyễn Thị Kim Phụng" ,
    "Mai Xuân Hùng"
  ] ,
  "HoTen": "Lê Thị Hồng Cúc" ,
  "Lop": "HTCL2019" ,
  "MaKhoa": "HTTT" ,
  "MaSV": "SV01" ,
  "TenKhoa": "Hệ Thống Thông tin"
}

```

2.4.6. Các hàm tổng hợp

So sánh giữa SQL và ReQL và các ví dụ:

SQL	ReQL
<pre>SELECT NuocSX FROM SanPham GROUP BY NuocSX</pre>	<pre>r.table("SanPham").map(r.row["NuocSX"]).distinct()</pre>
<pre>SELECT MaKH, NgayHD SUM('TriGia') FROM HoaDon GROUP BY MaKH, NgayHD</pre>	<pre>r.table('HoaDon') .group('MaKH', 'NgayHD') .sum('TriGia')</pre>
<pre>SELECT category, SUM(num_comments) FROM posts WHERE num_comments > 7 GROUP BY category</pre>	<pre>r.table("posts") .filter(r.row['num_comments']>7) .group('category') .sum('num_comments')</pre>
<pre>SELECT category, SUM(num_comments) FROM posts GROUP BY category HAVING num_comments > 7</pre>	<pre>r.table("posts") .group('category') .sum('num_comments') .ungroup() .filter(r.row["reduction"] > 7)</pre>

```
SELECT title,  
       COUNT(title)  
FROM Phim  
GROUP BY title  
HAVING COUNT(title) > 1
```

```
r.table("movies")  
  .group("title")  
  .count()  
  .ungroup()  
  .filter(r.row["reduction"] > 1)
```

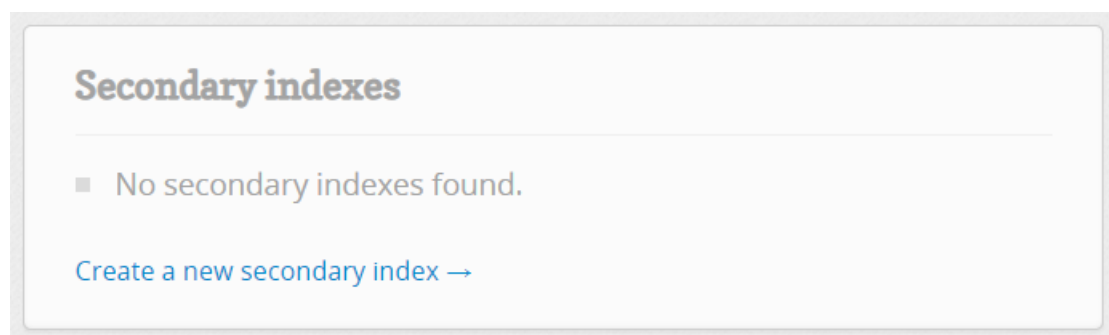
2.4.7. Đánh chỉ mục (Index)

Chỉ mục của mỗi bản ghi sẽ được RethinkDB sử dụng để đưa đến mảnh thích hợp. Và chỉ mục bên trong mỗi mảnh sẽ sử dụng cấu trúc dữ liệu B-Tree. Truy vấn sử dụng chỉ mục này sẽ rất hiệu quả bởi vì truy vấn sẽ được đưa ngay lập tức đến mảnh tương ứng và bản ghi sẽ được tìm kiếm nhanh chóng trên B-Tree.

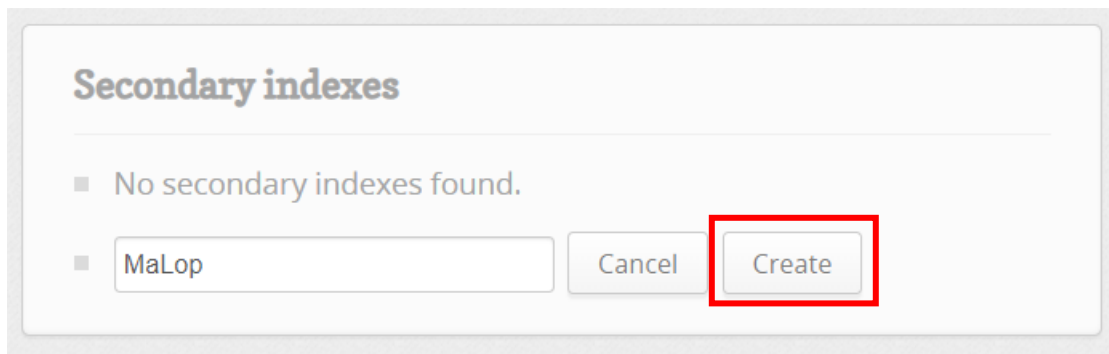
RethinkDB hỗ trợ cả chỉ mục thứ cấp và chỉ mục đa mức.

Cách thêm một chỉ mục cho một bảng:

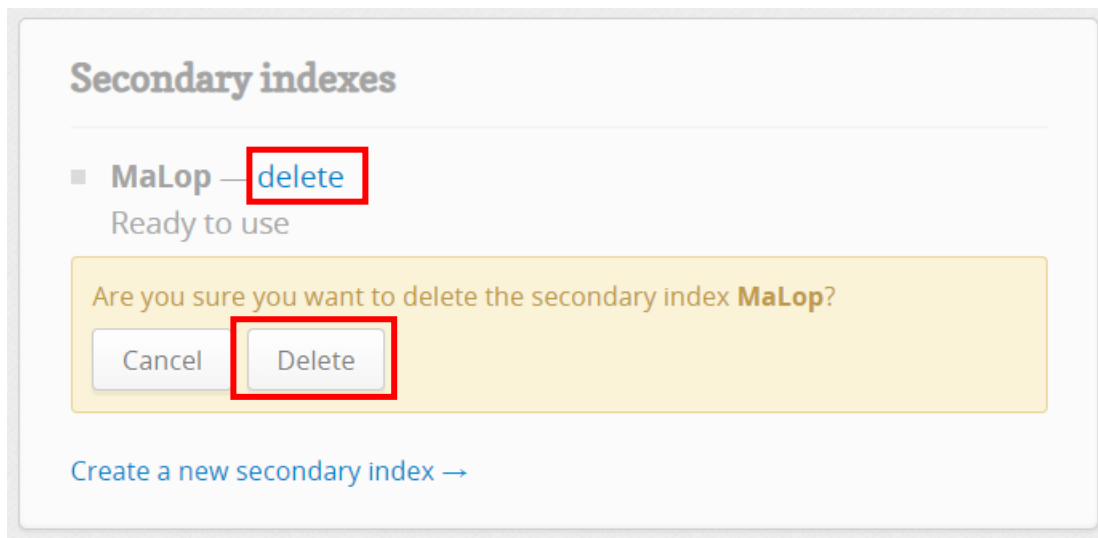
Bước 1: Chọn **"Create a new secondary index →"** tại mục Secondary indexes tại bảng cần tạo chỉ mục.



Bước 2: Tập tên chỉ mục và chọn Create để tạo.



Khi muốn xóa chỉ mục, ta button delete kế bên chỉ mục đó và chọn button delete phía dưới để xác nhận xóa chỉ mục.

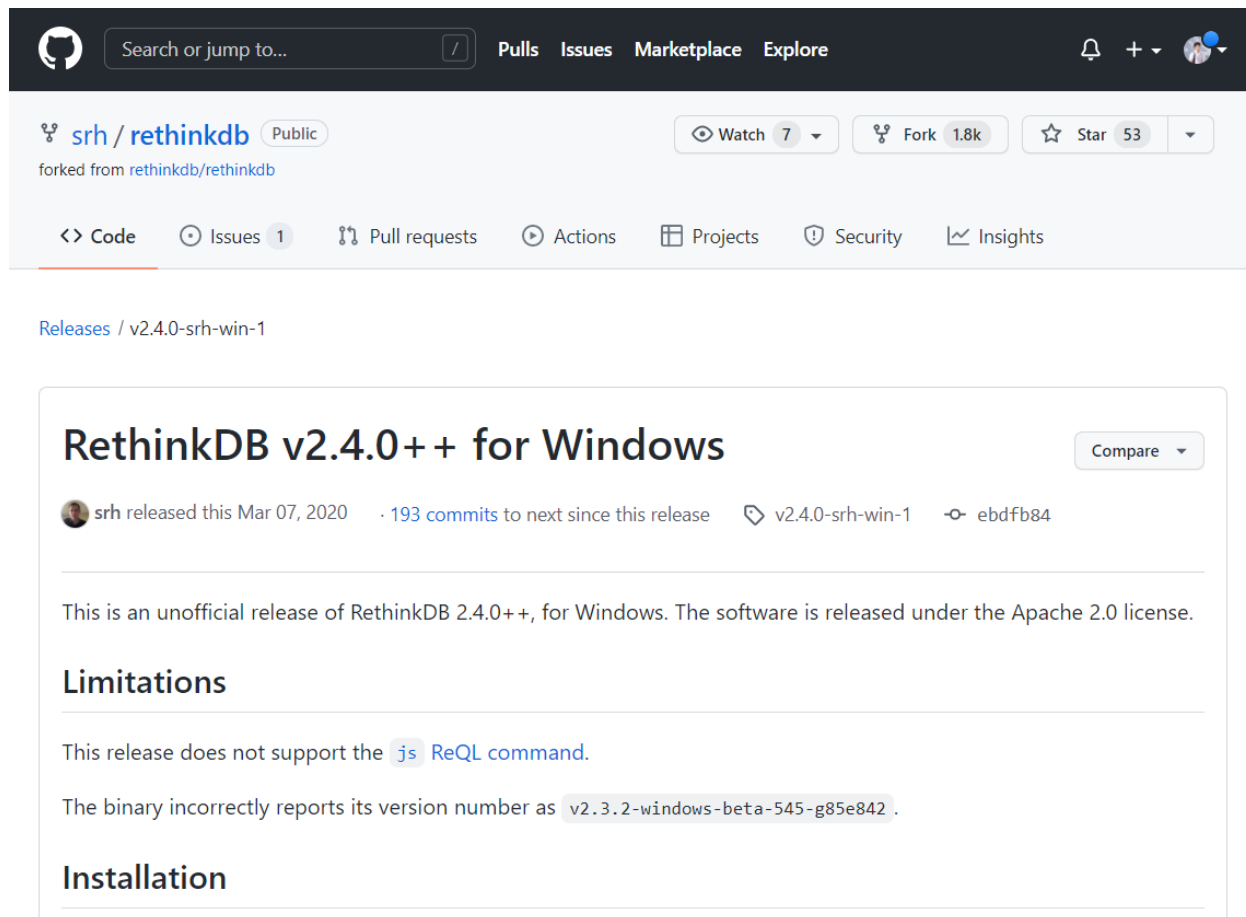


CHƯƠNG 2: CÀI ĐẶT TRÊN 2 MÁY TRỞ LÊN VÀ THỰC HIỆN TRUY VẤN GIỮA HAI MÁY

1. Cài đặt RethinkDB

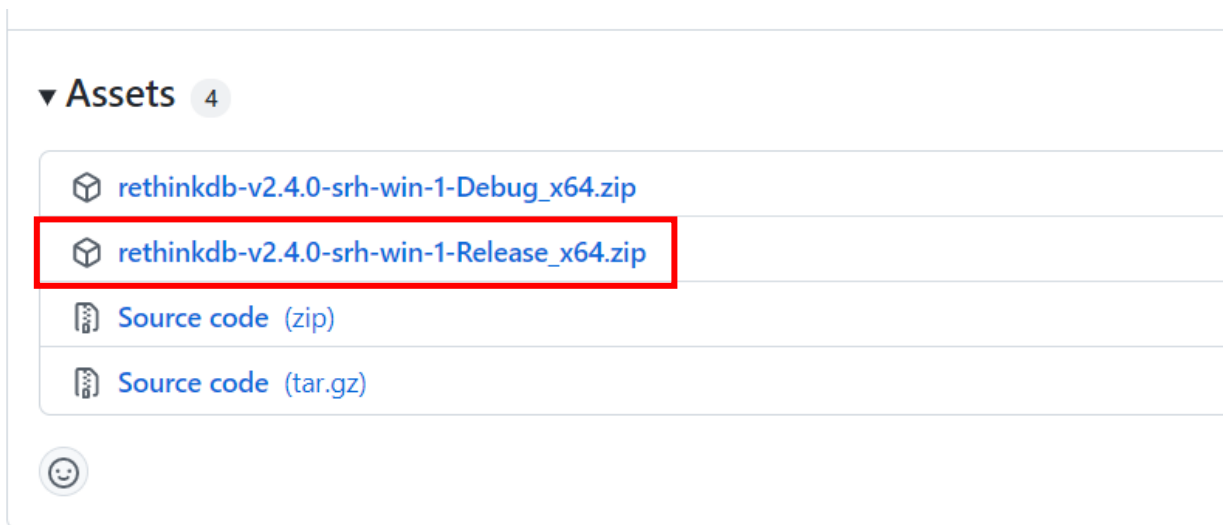
Bước 1: Chúng ta truy cập vào đường link sau:

[Release RethinkDB v2.4.0++ for Windows · srh/rethinkdb \(github.com\)](https://github.com/srh/rethinkdb/releases/tag/v2.4.0-srh-win-1)



The screenshot shows the GitHub interface for the repository `srh/rethinkdb`. The repository is public and has 7 watchers, 1.8k forks, and 53 stars. The navigation bar includes links for Code, Issues (1), Pull requests, Actions, Projects, Security, and Insights. The current view is the Releases page, specifically for the release `v2.4.0-srh-win-1`. The release title is **RethinkDB v2.4.0++ for Windows**. It was released on March 07, 2020, by user `srh`. The release description states: "This is an unofficial release of RethinkDB 2.4.0++, for Windows. The software is released under the Apache 2.0 license." Under the **Limitations** section, it notes: "This release does not support the `js ReQL command`." and "The binary incorrectly reports its version number as `v2.3.2-windows-beta-545-g85e842`." The **Installation** section is also visible.

Bước 2: Tại mục Assets, chọn file ***rethinkdb-v2.4.0-srh-win-1-Release_x64.zip*** tải về RethinkDB.



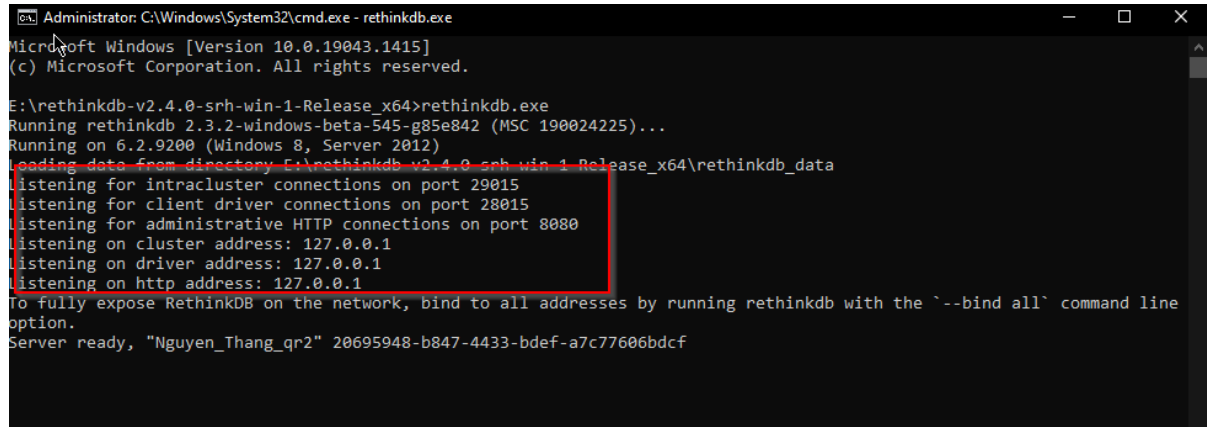
Bước 3: Giải nén file .zip vừa tải xuống

This PC > DATA (D:) > APPLICATION > rethinkdb-v2.4.0-srh-win-1-Release_x64				
Name	Date modified	Type	Size	
rethinkdb_data	12/19/2021 9:32 PM	File folder		
rethinkdb_data2	12/19/2021 9:15 PM	File folder		
rethinkdb_data3	12/19/2021 9:17 PM	File folder		
rethinkdb	3/7/2020 2:07 PM	Application	24,435 KB	
rethinkdb.pdb	3/7/2020 2:07 PM	Program Debug D...	158,220 KB	
vc140.pdb	3/7/2020 2:07 PM	Program Debug D...	62,940 KB	

Bước 4: Để khởi động RethinkDB ta dùng cmd hoặc terminal và trở đến thư mục RethinkDB vừa giải nén và chạy lệnh:

Rethinkdb.exe

Sau khi chạy lệnh trên xong, ta sẽ được kết quả như hình dưới



```
Administrator: C:\Windows\System32\cmd.exe - rethinkdb.exe
Microsoft Windows [Version 10.0.19043.1415]
(c) Microsoft Corporation. All rights reserved.

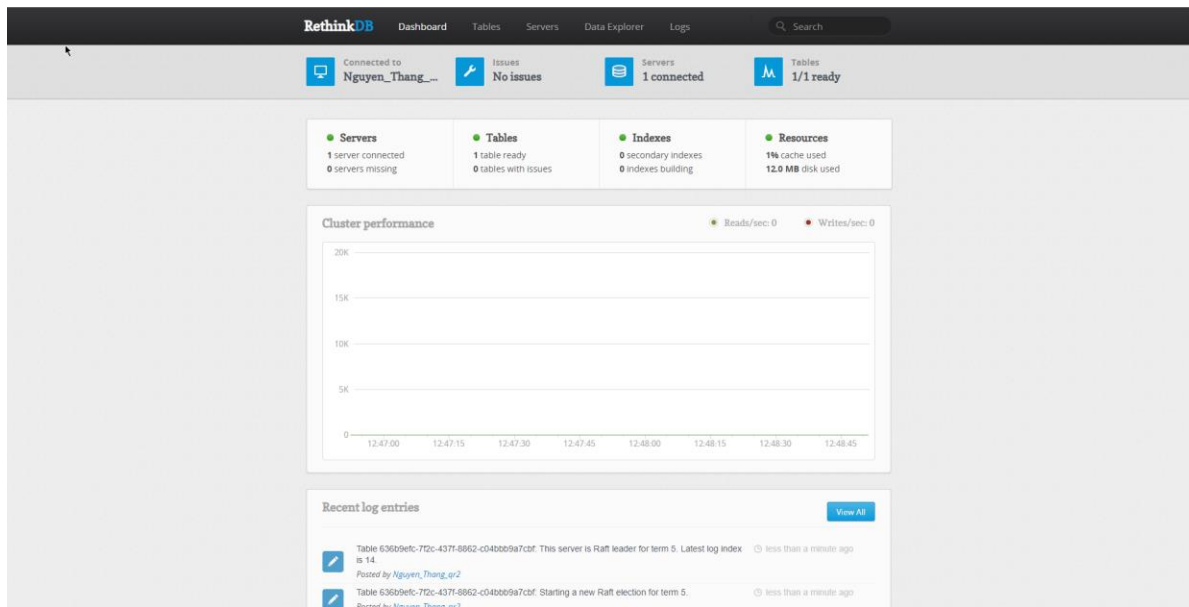
E:\rethinkdb-v2.4.0-srh-win-1-Release_x64>rethinkdb.exe
Running rethinkdb 2.3.2-windows-beta-545-g85e842 (MSC 190024225)...
Running on 6.2.9200 (Windows 8, Server 2012)
Loading data from directory E:\rethinkdb-v2.4.0-srh-win-1-Release_x64\rethinkdb_data
listening for intracluster connections on port 29015
listening for client driver connections on port 28015
listening for administrative HTTP connections on port 8080
listening on cluster address: 127.0.0.1
listening on driver address: 127.0.0.1
listening on http address: 127.0.0.1
To fully expose RethinkDB on the network, bind to all addresses by running rethinkdb with the '--bind all' command line option.
Server ready, "Nguyen_Thang_qr2" 20695948-b847-4433-bdef-a7c77606bdcf
```

Ở đây chúng ta có được 3 cổng port là intracluster, client driver và administrative HTTP

- Intracluster: là cổng để các máy server có thể kết nối được với nhau.
- Client driver: là cổng để các phần mềm chương trình có thể kết nối đến server.
- Administrative HTTP: là cổng để truy cập đến giao diện web của RethinkDB.

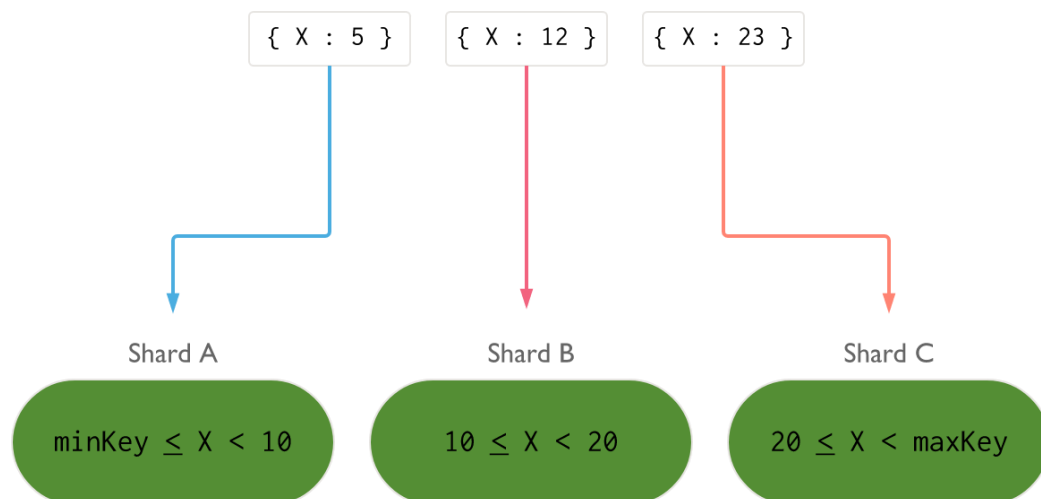
Tương ứng với các cổng port trên sẽ là địa chỉ của nó, ở đây địa chỉ mặc định sẽ là localhost (127.0.0.1)

Để truy cập đến giao diện web của RethinkDB, ta vào browser và nhập vào địa chỉ: `127.0.0.1:8080`

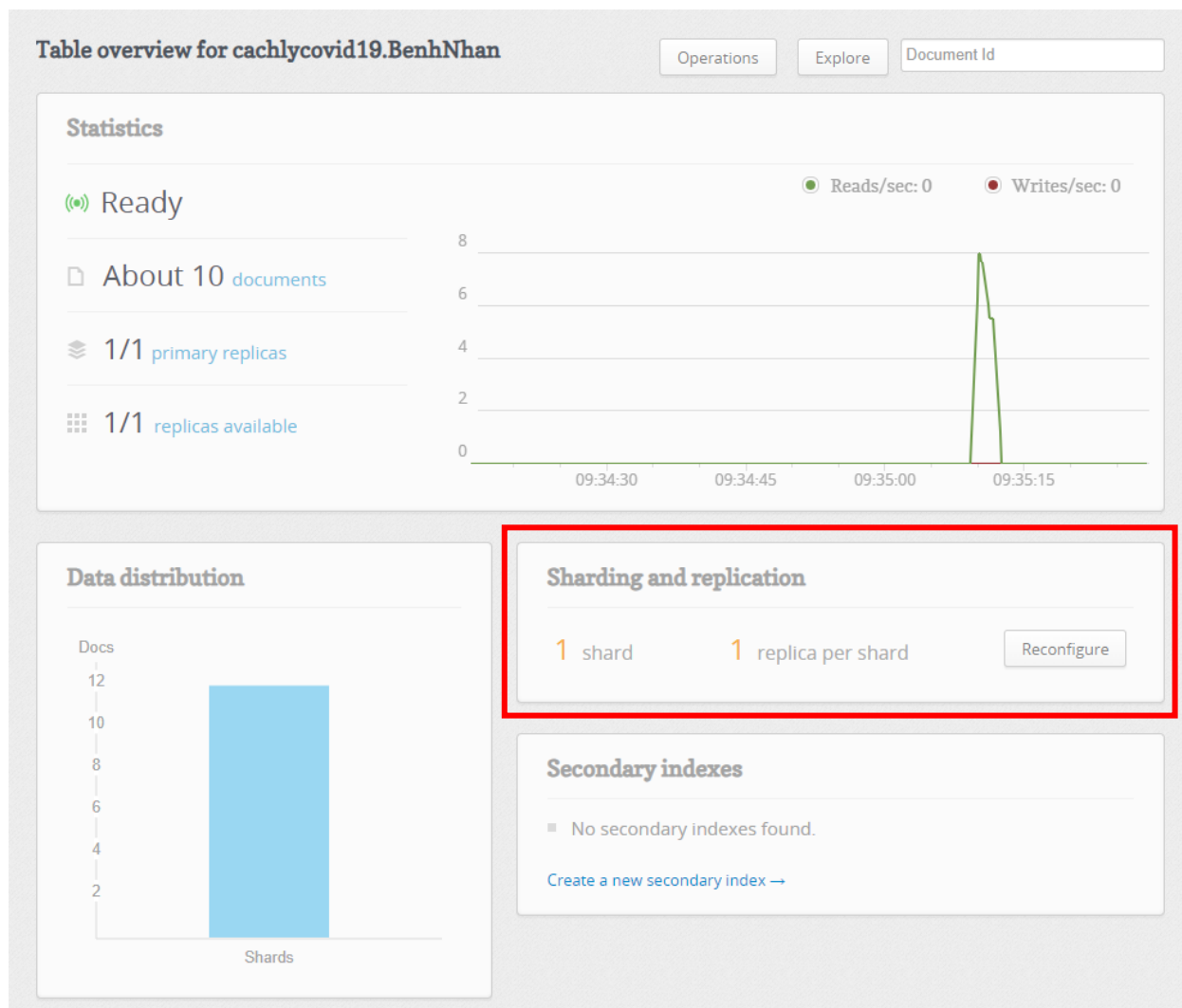


2. Cơ chế phân tán của RethinkDB

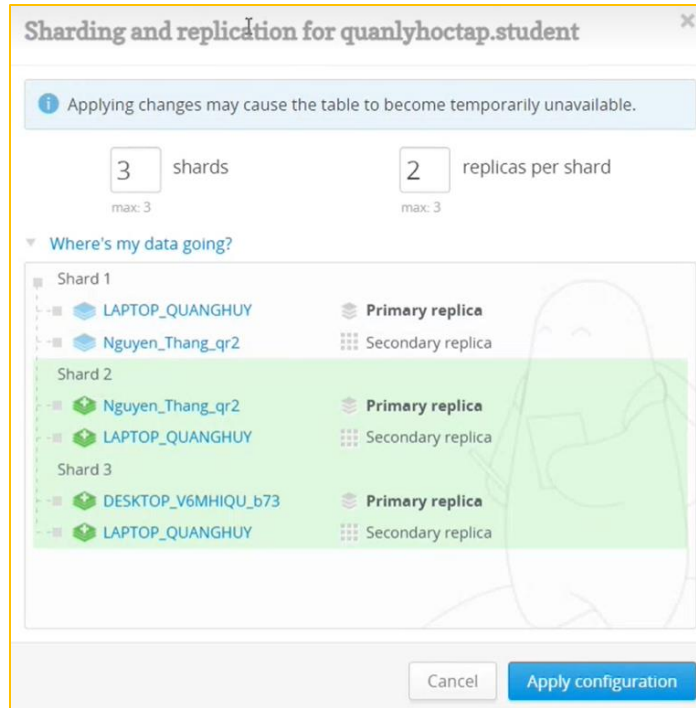
RethinkDB sử dụng thuật toán phân mảnh theo phạm vi (range sharding algorithm) dựa trên khóa chính của các bảng để phân vùng dữ liệu.



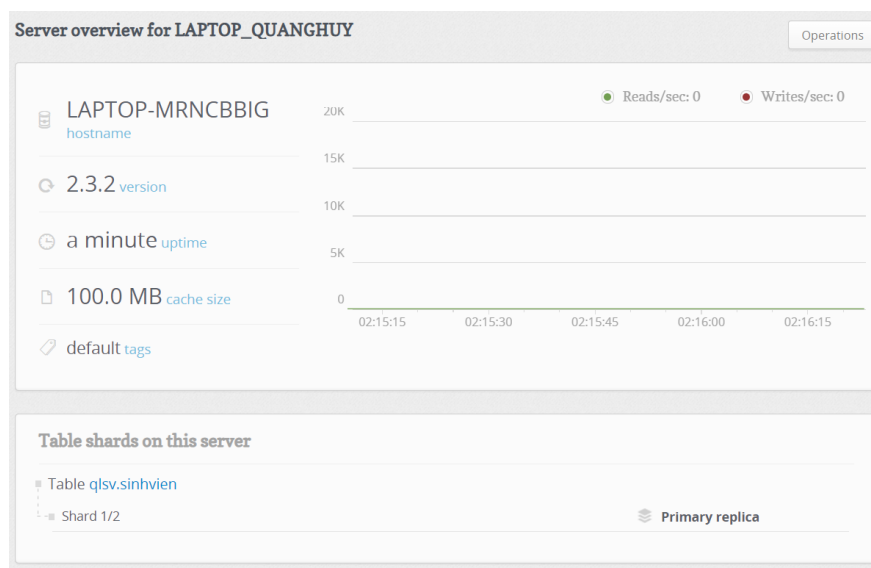
Dựa trên các điều kiện người dùng yêu cầu hệ thống sẽ kiểm tra các số liệu thống kê của bảng để đưa ra các điểm phân chia tối ưu cho bảng đó. Các điểm đó sẽ đảm bảo các mảnh sẽ chứa số lượng các bản ghi tương tự nhau và chúng sẽ được phân bố tự động trên toàn bộ cụm (cluster).



Các chức năng sharding và replication được cài đặt trong bảng cấu hình, chúng cho phép người dùng chỉ định số mảnh và số bản sao trên mỗi bảng hoặc tất cả các bảng bên trong cơ sở dữ liệu.

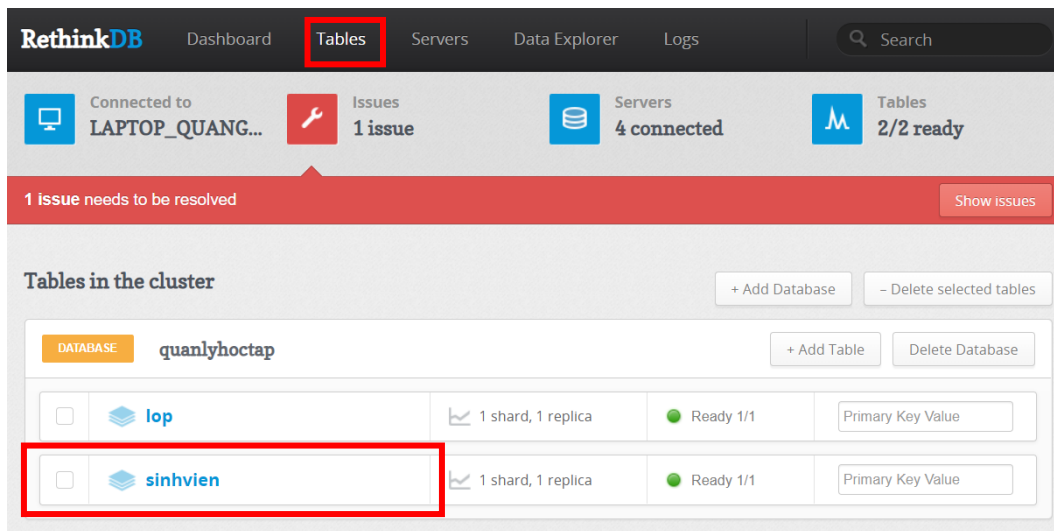


RethinkDB giữ một thư mục nội bộ theo dõi tình trạng hiện tại của cụm (cluster): có bao nhiêu máy chủ có thể truy cập được, dữ liệu nào được lưu trữ trên mỗi máy chủ,... Các cấu trúc dữ liệu theo dõi thư mục được tự động cập nhật khi cụm (cluster) thay đổi.



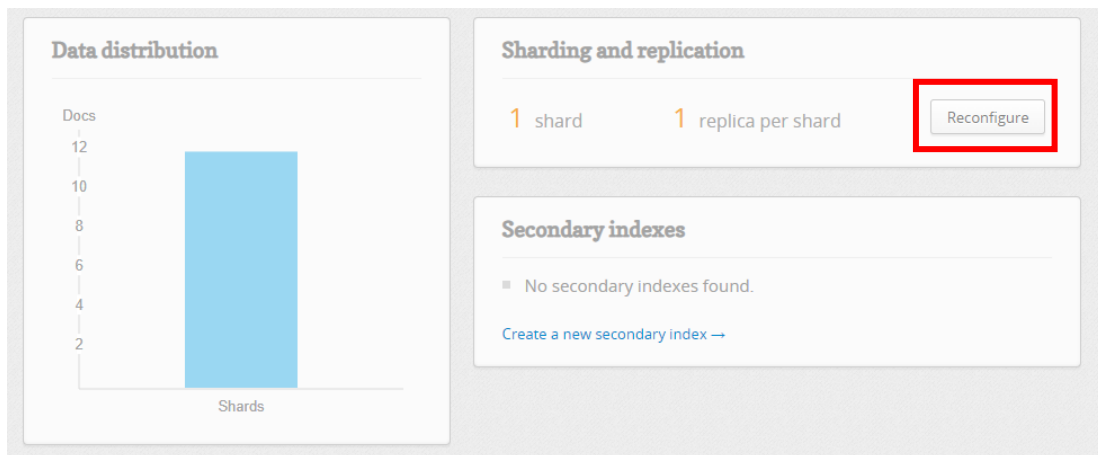
Khi sử dụng giao diện người dùng web của RethinkDB, chỉ cần chỉ định số lượng phân đoạn chúng ta muốn và dựa trên dữ liệu có sẵn, RethinkDB sẽ xác định các điểm phân tách tốt nhất để duy trì các phân đoạn cân bằng để chia nhỏ dữ liệu.

Bước 1: Chọn Tables và chọn bảng cần muốn phân mảnh



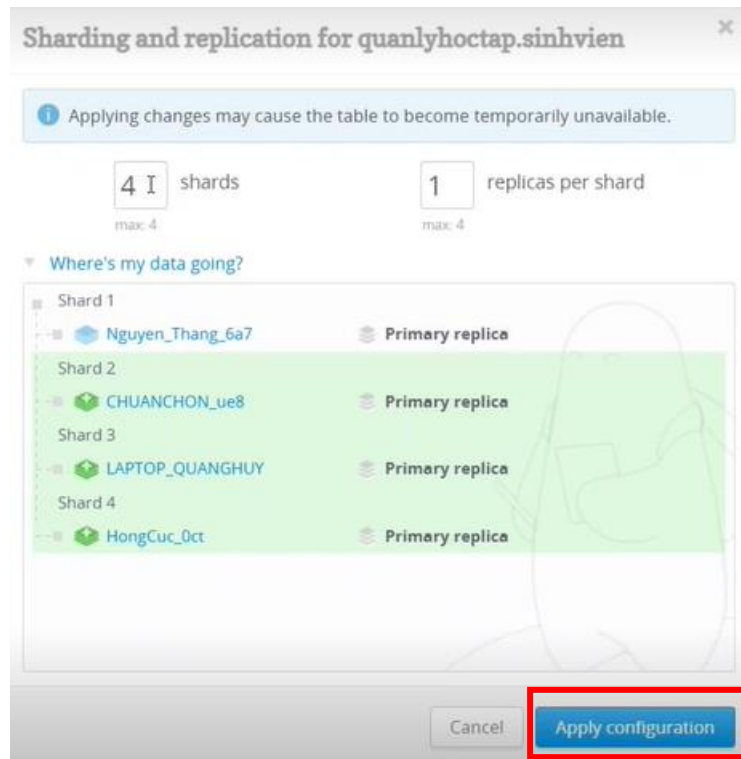
The screenshot shows the RethinkDB web interface. The 'Tables' tab is selected in the top navigation bar. Below the navigation bar, there are status indicators: 'Connected to LAPTOP_QUANG...', 'Issues 1 issue', 'Servers 4 connected', and 'Tables 2/2 ready'. A red banner indicates '1 issue needs to be resolved'. The 'Tables in the cluster' section shows a table named 'quanlyhochtap' with two sub-tables: 'lop' and 'sinhvien'. The 'sinhvien' table is highlighted with a red box. It shows '1 shard, 1 replica' and 'Ready 1/1' status.

Bước 2: Nhấn vào nút Reconfigure.



The screenshot shows the 'Sharding and replication' section of the RethinkDB web interface. It displays '1 shard' and '1 replica per shard'. The 'Reconfigure' button is highlighted with a red box. Below this, the 'Secondary indexes' section shows 'No secondary indexes found' and a link to 'Create a new secondary index →'.

Bước 3: Chọn số lượng mảnh và bản sao cần phân mảnh. Trước khi cập nhật cấu hình, hãy nhấp vào mục *Where's my data going?* để xem dữ liệu sẽ được lan truyền như thế nào. Chẳng hạn như ví dụ này hiển thị 4 phân mảnh với 1 bản sao:



Lưu ý cách các bản sao được trải ra ở nhiều nơi khác nhau - nếu có điều gì đó xảy ra trên một nút, các bản sao khác của chúng ta vẫn an toàn và bình thường.

Bước 4: Nhấn nút Apply configuration để áp dụng cấu hình.

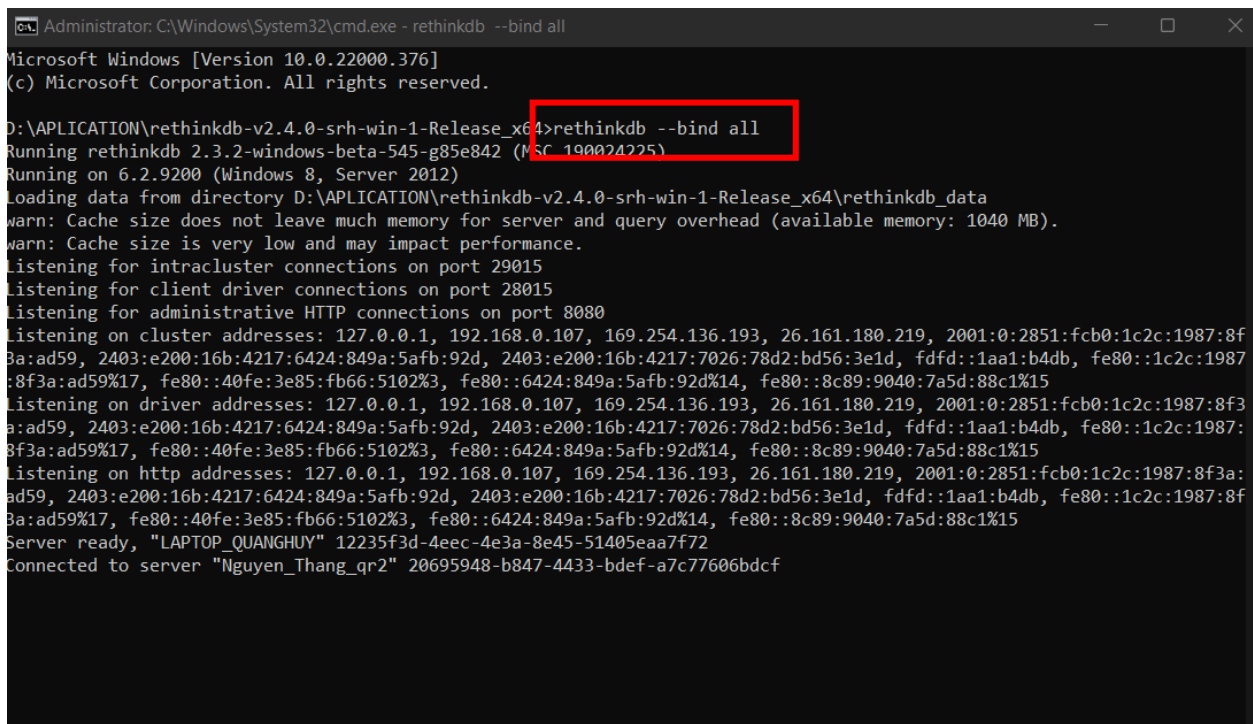
3. Kết nối nhiều máy thông qua mạng LAN ảo

Ở đây, chúng ta sẽ thực hiện kết nối các server (các máy) lại với nhau bằng cách sử dụng phần mềm ảo hóa mạng LAN: RadminVPN.

Có lẽ chúng ta không còn quá xa lạ gì với RadminVPN nữa nên phần này mình không nói chi tiết về nó.

Bước 1: Hãy đảm bảo rằng các máy đã có chung network RadminVPN và việc “ping” qua lại giữa các máy diễn ra tốt đẹp.

Bước 2: Máy chủ chạy lệnh `rethinkdb --bind all` để chạy server máy chủ của mình trên mạng LAN ảo



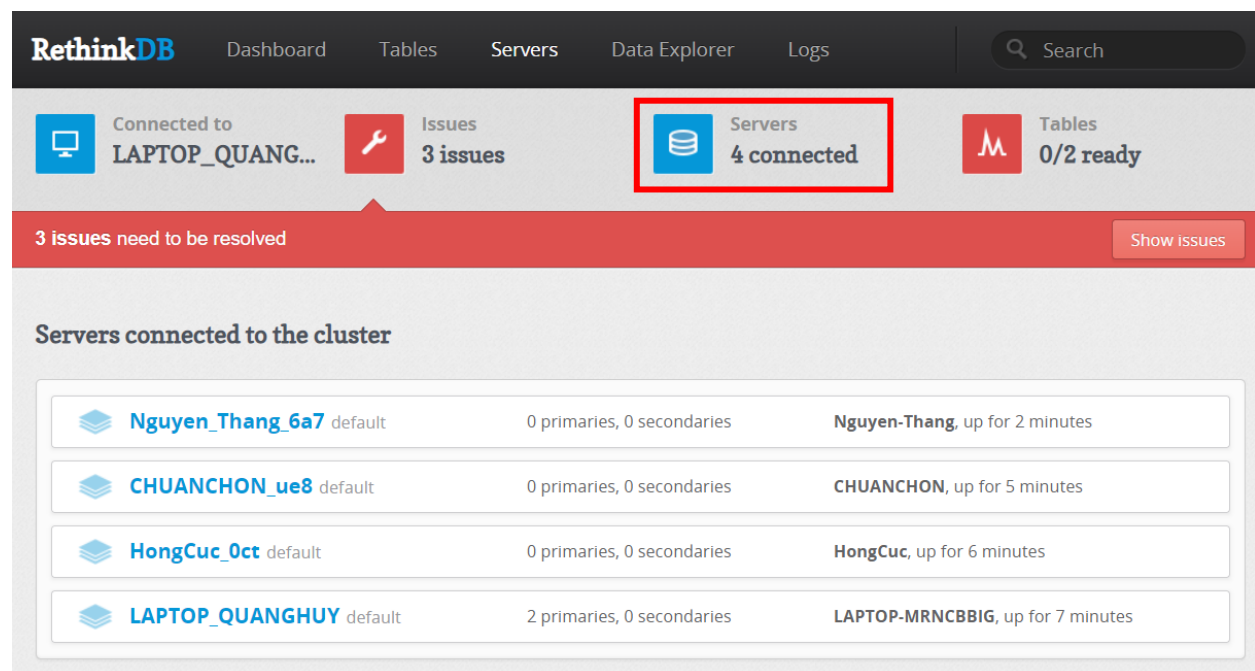
```
Administrator: C:\Windows\System32\cmd.exe - rethinkdb --bind all
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

D:\APPLICATION\rethinkdb-v2.4.0-srh-win-1-Release_x64>rethinkdb --bind all
Running rethinkdb 2.3.2-windows-beta-545-g85e842 (MSC 190024225)
Running on 6.2.9200 (Windows 8, Server 2012)
Loading data from directory D:\APPLICATION\rethinkdb-v2.4.0-srh-win-1-Release_x64\rethinkdb_data
warn: Cache size does not leave much memory for server and query overhead (available memory: 1040 MB).
warn: Cache size is very low and may impact performance.
Listening for intracluster connections on port 29015
Listening for client driver connections on port 28015
Listening for administrative HTTP connections on port 8080
Listening on cluster addresses: 127.0.0.1, 192.168.0.107, 169.254.136.193, 26.161.180.219, 2001:0:2851:fc0:1c2c:1987:8f3a:ad59, 2403:e200:16b:4217:6424:849a:5afb:92d, 2403:e200:16b:4217:7026:78d2:bd56:3e1d, fdff::1aa1:b4db, fe80::1c2c:1987:8f3a:ad59%17, fe80::40fe:3e85:fb66:5102%3, fe80::6424:849a:5afb:92d%14, fe80::8c89:9040:7a5d:88c1%15
Listening on driver addresses: 127.0.0.1, 192.168.0.107, 169.254.136.193, 26.161.180.219, 2001:0:2851:fc0:1c2c:1987:8f3a:ad59, 2403:e200:16b:4217:6424:849a:5afb:92d, 2403:e200:16b:4217:7026:78d2:bd56:3e1d, fdff::1aa1:b4db, fe80::1c2c:1987:8f3a:ad59%17, fe80::40fe:3e85:fb66:5102%3, fe80::6424:849a:5afb:92d%14, fe80::8c89:9040:7a5d:88c1%15
Listening on http addresses: 127.0.0.1, 192.168.0.107, 169.254.136.193, 26.161.180.219, 2001:0:2851:fc0:1c2c:1987:8f3a:ad59, 2403:e200:16b:4217:6424:849a:5afb:92d, 2403:e200:16b:4217:7026:78d2:bd56:3e1d, fdff::1aa1:b4db, fe80::1c2c:1987:8f3a:ad59%17, fe80::40fe:3e85:fb66:5102%3, fe80::6424:849a:5afb:92d%14, fe80::8c89:9040:7a5d:88c1%15
Server ready, "LAPTOP_QUANGHUY" 12235f3d-4eec-4e3a-8e45-51405eaa7f72
Connected to server "Nguyen_Thang_qr2" 20695948-b847-4433-bdef-a7c77606bdcf
```


Bước 3: Các máy trạm chạy lệnh `rethinkdb.exe --directory rethinkdb_data2 --join <ĐỊA CHỈ IP MÁY CHỦ>`

```
F:\MyUniversityIT\Year03\CSDLPhanTan\rethinkdb-v2.4.0-srh-win-1-Release_x64\rethinkdb.exe --directory rethinkdb_data2 --join 26.161.180.219
Running rethinkdb 2.3.2-windows-beta-545-g85e842 (MSC 190024225)...
Running on 6.2.9200 (Windows 8, Server 2012)
Loading data from directory F:\MyUniversityIT\Year03\CSDLPhanTan\rethinkdb-v2.4.0-srh-win-1-Release_x64\rethinkdb_data2
Listening for intracluster connections on port 29015
Listening for client driver connections on port 28015
Listening for administrative HTTP connections on port 8080
Listening on cluster address: 127.0.0.1
Listening on driver address: 127.0.0.1
Listening on http address: 127.0.0.1
To fully expose RethinkDB on the network, bind to all addresses by running rethinkdb with the `--bind all` command line option.
Server ready, "HongCuc_0ct" 5616d4c2-8b80-4daf-aeb1-4ecd1e37e8be
Connected to server "LAPTOP_QUANGHUY" 12235f3d-4eec-4e3a-8e45-51405eaa7f72
```

Bước 4: Vào tab Servers của RethinkDB để xem thông tin Server đang kết nối với nhau.



Server Name	Role	Primary/Secondary Count	Status
Nguyen_Thang_6a7	default	0 primaries, 0 secondaries	Nguyen-Thang, up for 2 minutes
CHUANCHON_ue8	default	0 primaries, 0 secondaries	CHUANCHON, up for 5 minutes
HongCuc_0ct	default	0 primaries, 0 secondaries	HongCuc, up for 6 minutes
LAPTOP_QUANGHUY	default	2 primaries, 0 secondaries	LAPTOP-MRNCBBIG, up for 7 minutes

CHƯƠNG 3: THAO TÁC DỮ LIỆU QUA NHIỀU MÁY

1. Thao tác với RethinkDB

CREATE TABLE MÁY 1

```
r.db('COVID19').tableCreate('BenhNhan', {primaryKey: "MaBN"})
r.db('COVID19').tableCreate('DiemCachLy', {primaryKey: "MaDCL"})
r.db('COVID19').tableCreate('NhanVienDieuTri', {primaryKey: "MaNV"})
```

INSERT DATA MÁY 1

```
-- Table DiemCachLy

r.db('COVID19').table("DiemCachLy").insert(
  [
    {
      "MaDCL": "CL01",
      "TenDiemCachLy": "Điểm cách ly TP.Dĩ An",
      "DiaChi": "Dĩ An, Đông Hòa, Bình Dương"
    },
    {
      "MaDCL": "CL02",
      "TenDiemCachLy": "Điểm cách ly TP.Biên Hòa",
      "DiaChi": "30 Nguyễn Trãi, TP.Biên Hòa - Đồng Nai"
    },
    {
      "MaDCL": "CL03",
      "TenDiemCachLy": "Điểm cách ly TP.Thủ Đức",
      "DiaChi": "30 Võ Văn Ngân, TP.Thủ Đức"
    }
  ]
)
```

```

    },
    {
        "MaDCL": "CL04",
        "TenDiemCachLy": "Điểm cách ly huyện Củ Chi",
        "DiaChi": "ấp 4, huyện Củ Chi, TP HCM"
    }
]
);

```

```
-- Table NhanVienDieuTri
```

```

r.db('COVID19').table("NhanVienDieuTri").insert(
[
    {
        "MaNV": "NV01",
        "HoTen": "Phạm Văn Tuấn",
        "GioiTinh": "Nam",
        "DonVi": "CL01"
    },
    {
        "MaNV": "NV02",
        "HoTen": "Nguyễn Tuấn Anh",
        "GioiTinh": "Nam",
        "DonVi": "CL02"
    },
    {
        "MaNV": "NV03",
        "HoTen": "Hoàng Thị Hoài Anh",
        "GioiTinh": "Nữ",
        "DonVi": "CL02"
    },
    {

```

```
{
  "MaNV": "NV04",
  "HoTen": "Trần Đại Nam",
  "GioiTinh": "Nam",
  "DonVi": "CL01"
},
{
  "MaNV": "NV05",
  "HoTen": "Nguyễn Ngọc An",
  "GioiTinh": "Nữ",
  "DonVi": "CL01"
},
{
  "MaNV": "NV06",
  "HoTen": "Nguyễn Lê Thảo Duyên",
  "GioiTinh": "Nữ",
  "DonVi": "CL02"
},
{
  "MaNV": "NV07",
  "HoTen": "Lê Quốc Tín",
  "GioiTinh": "NNam",
  "DonVi": "CL02"
},
{
  "MaNV": "NV08",
  "HoTen": "Phạm Hoàng Mỹ",
  "GioiTinh": "Nữ",
  "DonVi": "CL01"
},
{
```

```

        "MaNV": "NV09",
        "HoTen": "Trần Nhật Tân",
        "GioiTinh": "Nam",
        "DonVi": "CL01"
    },
    {
        "MaNV": "NV10",
        "HoTen": "Phạm Thu Giang",
        "GioiTinh": "Nữ",
        "DonVi": "CL02"
    }
]);

--Table BenhNhan

r.db('COVID19').table("BenhNhan").insert(
[
    {
        "MaBN": "BN01",
        "HoTen": "Nguyễn Văn Hiếu",
        "TrieuChung": [
            "Ho", "Sốt", "Khó thở"
        ],
        "GioiTinh": "Nam",
        "NuocVe": "Trung Quốc",
        "MaDCL": "CL01",
        "NhanVienDieuTri": ["NV01", "NV03"]
    },
    {
        "MaBN": "BN02",
        "HoTen": "Lê Kim Duyên",

```

```

    "TrieuChung": [
        "Ho", "Sốt", "Khó thở"
    ],
    "GioiTinh": "Nữ",
    "NuocVe": "Hàn Quốc",
    "MaDCL": "CL02",
    "NhanVienDieuTri": ["NV02", "NV04"]
    },
{
    "MaBN": "BN03",
    "HoTen": "Nguyễn Thị Trâm",
    "TrieuChung": [
        "Tức ngực", "Khó thở"
    ],
    "GioiTinh": "Nữ",
    "NuocVe": "Anh",
    "MaDCL": "CL01",
    "NhanVienDieuTri": ["NV05", "NV07"]
    },
{
    "MaBN": "BN04",
    "HoTen": "Hoàng Văn Tiến Đạt",
    "TrieuChung": [
        "Sốt", "Tức ngực"
    ],
    "GioiTinh": "Nam",
    "NuocVe": "Nhật Bản",
    "MaDCL": "CL02",
    "NhanVienDieuTri": ["NV06", "NV08"]
},
{

```

```

"MaBN": "BN05",
"HoTen": "Đỗ Văn Nam",
"TrieuChung": [
    "Tức ngực", "Khó thở", "Ho", "Sốt"
],
"GioiTinh": "Nam",
"NuocVe": "Canada",
"MaDCL": "CL01",
"NhanVienDieuTri": ["NV09", "NV10"]
},
{
"MaBN": "BN06",
"HoTen": "Trần Văn Minh",
"TrieuChung": [
    "Khó thở", "Sốt"
],
"GioiTinh": "Nam",
"NuocVe": "Mỹ",
"MaDCL": "CL01",
"NhanVienDieuTri": ["NV01", "NV03"]
},
{
"MaBN": "BN07",
"HoTen": "Nguyễn Trần Khánh Vân",
"TrieuChung": [
    "Tức ngực", "Khó thở"
],
"GioiTinh": "Nữ",
"NuocVe": "Singapore",
"MaDCL": "CL02",
"NhanVienDieuTri": ["NV02", "NV04"]
}

```

```

    },
{
    "MaBN": "BN08",
    "HoTen": "Phan Văn Tài",
    "TrieuChung": [
        "Ho", "Sốt"
    ],
    "GioiTinh": "Nam",
    "NuocVe": "Trung Quốc",
    "MaDCL": "CL01",
    "NhanVienDieuTri": ["NV05", "NV07"]
    },
{
    "MaBN": "BN09",
    "HoTen": "Nguyễn Minh An",
    "TrieuChung": [
        "Sốt"
    ],
    "GioiTinh": "Nam",
    "NuocVe": "Mỹ",
    "MaDCL": "CL02",
    "NhanVienDieuTri": ["NV06", "NV08"]
    },
{
    "MaBN": "BN10",
    "HoTen": "Nguyễn Tấn Khang",
    "TrieuChung": [
        "Tức ngực", "Khó thở", "Ho", "Sốt"
    ],
    "GioiTinh": "Nam",
    "NuocVe": "Anh",

```



```

        "MaDCL": "CL01",
        "NhanVienDieuTri": ["NV09", "NV10"]
    }
]);

```

CREATE TABLE MÁY 2

```

r.db('COVID19_2').tableCreate('BenhNhan', {primaryKey: "MaBN"})
r.db('COVID19_2').tableCreate('DiemCachLy', {primaryKey: "MaDCL"})
r.db('COVID19_2').tableCreate('NhanVienDieuTri', {primaryKey: "MaNV"})

```

INSERT DATA MÁY 2

```

r.db('COVID19_2').tableCreate('DiemCachLy', {primaryKey: "MaDCL"})

r.db('COVID19_2').tableCreate('NhanVienDieuTri', {primaryKey: "MaNV"})

r.db('COVID19_2').tableCreate('BenhNhan', {primaryKey: "MaBN"})

r.db('COVID19_2').table("BenhNhan").insert([
{
    "MaBN": "BN11",
    "HoTen": "Nguyễn Thị Thủy",
    "TrieuChung": [
        "Khó thở", "Ho"
    ],
    "GioiTinh": "Nữ",
    "NuocVe": "Singapore",

```

```

"MaDCL": "CL03",
"NhanVienDieuTri": ["NV11", "NV12"],
},
{
"MaBN": "BN12",
"HoTen": "Trần Văn Tiến",
"TrieuChung": [
"Ho"
],
"GioiTinh": "Nam",
"NuocVe": "Campuchia",
"MaDCL": "CL03",
"NhanVienDieuTri": ["NV11", "NV13"],
},
{
"MaBN": "BN14",
"HoTen": "Lê Kim Duyên",
"TrieuChung": [
"Ho", "Sốt", "Khó thở"
],
"NuocVe": "Anh",
"GioiTinh": "Nữ",
"MaDCL": "CL03",
"NhanVienDieuTri": ["NV12", "NV13"],
},
{
"MaBN": "BN13",
"HoTen": "Nguyễn Phan Gia Hân",
"TrieuChung": [
"Ho", "Sốt", "Khó thở", "Nhức đầu", "Mất vị giác"
],

```

```

        "NuocVe": "Trung Quoc",
        "GioiTinh": "Nữ",
        "MaDCL": "CL03",
        "NhanVienDieuTri": ["NV14", "NV15"],
    },
    {
        "MaBN": "BN15",
        "HoTen": "Nguyễn Diệp Chí Báo",
        "TrieuChung": [
            "Ho", "Khó thở", "Tức ngực"
        ],
        "NuocVe": "Canada",
        "GioiTinh": "Nam",
        "MaDCL": "CL03",
        "NhanVienDieuTri": ["NV14", "NV15"],
    },
    {
        "MaBN": "BN16",
        "HoTen": "Võ Nguyễn Hoài Linh",
        "TrieuChung": [
            "Ho", "Sốt", "Khó thở"
        ],
        "NuocVe": "Trung Quoc",
        "GioiTinh": "Nam",
        "MaDCL": "CL04",
        "NhanVienDieuTri": ["NV16"],
    },
    {
        "MaBN": "BN17",
        "HoTen": "Lê Trần Kim Anh",
        "TrieuChung": [

```

```

        "Mất vị giác", "Sốt", "Khó thở"
    ],
    "NuocVe": "Anh",
    "GioiTinh": "Nữ",
    "MaDCL": "CL04",
    "NhanVienDieuTri": ["NV16", "NV18"],
    },
    {
        "MaBN": "BN18",
        "HoTen": "Nguyễn Văn Hoài Tử",
        "TrieuChung": [
            "Ho", "Sốt", "Khó thở"
        ],
        "NuocVe": "Canada",
        "GioiTinh": "Nam",
        "MaDCL": "CL04",
        "NhanVienDieuTri": ["NV17", "NV18"],
        },
    {
        "MaBN": "BN19",
        "HoTen": "Lữ Hành Thái Sơn",
        "TrieuChung": [
            "Tức ngực", "Mất vị giác"
        ],
        "NuocVe": "Trung Quốc",
        "GioiTinh": "Nam",
        "MaDCL": "CL04",
        "NhanVienDieuTri": ["NV17"],
        },
    {
        "MaBN": "BN20",

```

```

        "HoTen": "Ngô Bá Khá",
        "TrieuChung": [
            "Ho", "Tức ngực", "Khó thở"
        ],
        "NuocVe": "Tây Ban Nha",
        "GioiTinh": "Nam",
        "MaDCL": "CL04",
        "NhanVienDieuTri": ["NV19", "NV20"],
    },
])

r.db('COVID19_2').table("NhanVienDieuTri").insert( [
{
    "MaNV": "NV11",
    "HoTen": "Nguyễn Hỏa Long Hoàng",
    "DonVi": "CL03",
    "GioiTinh": "Nam"
},
{
    "MaNV": "NV12",
    "HoTen": "Nguyễn Vàng Ánh Sao",
    "DonVi": "CL03",
    "GioiTinh": "Nữ"
},
{
    "MaNV": "NV13",
    "HoTen": "Dương La Nam Tự",
    "DonVi": "CL03",
    "GioiTinh": "Nam"
},
{

```

```
"MaNV": "NV14",
"HoTen": "Châu Khúc Tam Tình",
"DonVi": "CL03",
"GioiTinh": "Nữ"
},
{
"MaNV": "NV15",
"HoTen": "Đỗ Nhất Bách Tam",
"DonVi": "CL03",
"GioiTinh": "Nam"
},
{
"MaNV": "NV16",
"HoTen": "Lê Dương Hồng Trần",
"DonVi": "CL04",
"GioiTinh": "Nữ"
},
{
"MaNV": "NV17",
"HoTen": "Võ Ngất Bách",
"DonVi": "CL04",
"GioiTinh": "Nam"
},
{
"MaNV": "NV18",
"HoTen": "Cao Long Thị Cực",
"DonVi": "CL04",
"GioiTinh": "Nữ"
},
{
"MaNV": "NV19",
```

```

"HoTen": "Ngô La Thiện Trí",
"DonVi": "CL04",
"GioiTinh": "Nam"
},
{
"MaNV": "NV20",
"HoTen": "Hạ Hồ Thiên Giới",
"DonVi": "CL04",
"GioiTinh": "Nữ"
},
]);

```

QUERY

MÁY 1

SELECT

1. Cho biết những bệnh nhân có triệu chứng ho và sốt ở tất cả các điểm cách ly.

```

r.db('COVID19').table('BenhNhan').filter(r.row('TrieuChung').contains(
'Ho','Sốt')).union(r.db('COVID19_2').table('BenhNhan').filter(function
(item){
    return item('TrieuChung').contains('Ho','Sốt')
})))

```

2. Cho biết những bệnh nhân có nhiều hơn 3 triệu chứng và đang điều trị tại Dĩ An.

```
r.db('COVID19').table("BenhNhan").eqJoin("MaDCL",  
r.db('COVID19').table("DiemCachLy")).filter(function(benhnhan){  
return  
benhnhan('left')('TrieuChung').count().gt(3).and(benhnhan('right')('TenDiemCachLy').eq('Điểm cách ly TP.Dĩ An'))  
})
```

UPDATE

3. Cập nhật triệu chứng Mất vị giác, Ho, Tức ngực cho bệnh nhân có mã là BN06.

```
r.db('COVID19').table('BenhNhan').filter(r.row('MaBN').eq("BN06")).update({"TrieuChung":["Mất vị giác", "Ho", "Tức Ngực"]})
```

4. Cập nhật bệnh nhân có mã là BN05 được điều trị bởi nhân viên có mã là NV02.

```
r.db('COVID19').table('BenhNhan').filter(r.row('MaBN').eq("BN05")).update({"NhanVienDieuTri":["NV02"]})
```


DELETE
5. Xóa bệnh nhân có mã là BN03
<pre>r.db('COVID19').table('BenhNhan').filter(r.row('MaBN').eq("BN03")).delete()</pre>
6. Xóa những bệnh nhân được điều trị bởi nhân viên có mã NV01.
<pre>r.db('COVID19').table('BenhNhan').filter(r.row("NhanVienDieuTri").contains('NV01')).delete()</pre>

MÁY 2
SELECT
1. Cho biết những bệnh nhân có triệu chứng ho và sốt ở tất cả các điểm cách ly.
<pre>r.db('COVID19').table('BenhNhan').filter(r.row('TrieuChung').contains('Ho','Sốt')).union(r.db('COVID19_2').table('BenhNhan').filter(function(item){ return item('TrieuChung').contains('Ho','Sốt') })))</pre>

2. Cho biết những bệnh nhân có nhiều hơn 3 triệu chứng và đang điều trị tại TP. Thủ Đức.

```
r.db('COVID19_2').table("BenhNhan").eqJoin("MaDCL",  
r.db('COVID19_2').table("DiemCachLy")).filter(function(benhnhan){  
return  
benhnhan('left')('TrieuChung').count().gt(3).and(benhnhan('right')('TenDiemCachLy').eq('Điểm cách ly TP.Thủ Đức'))  
})
```

UPDATE

3. Cập nhật triệu chứng Mất vị giác, Ho, Tức ngực cho bệnh nhân có mã là BN13.

```
r.db('COVID19_2').table('BenhNhan').filter(r.row('MaBN').eq("BN13")).update({"TrieuChung":["Mất vị giác","Ho", "Tức Ngực"]})
```

4. Cập nhật bệnh nhân có mã là BN15 được điều trị bởi nhân viên có mã là NV11.

```
r.db('COVID19_2').table('BenhNhan').filter(r.row('MaBN').eq("BN15")).update({"NhanVienDieuTri":["NV11"]})
```

DELETE

5. Xóa bệnh nhân có mã là BN15

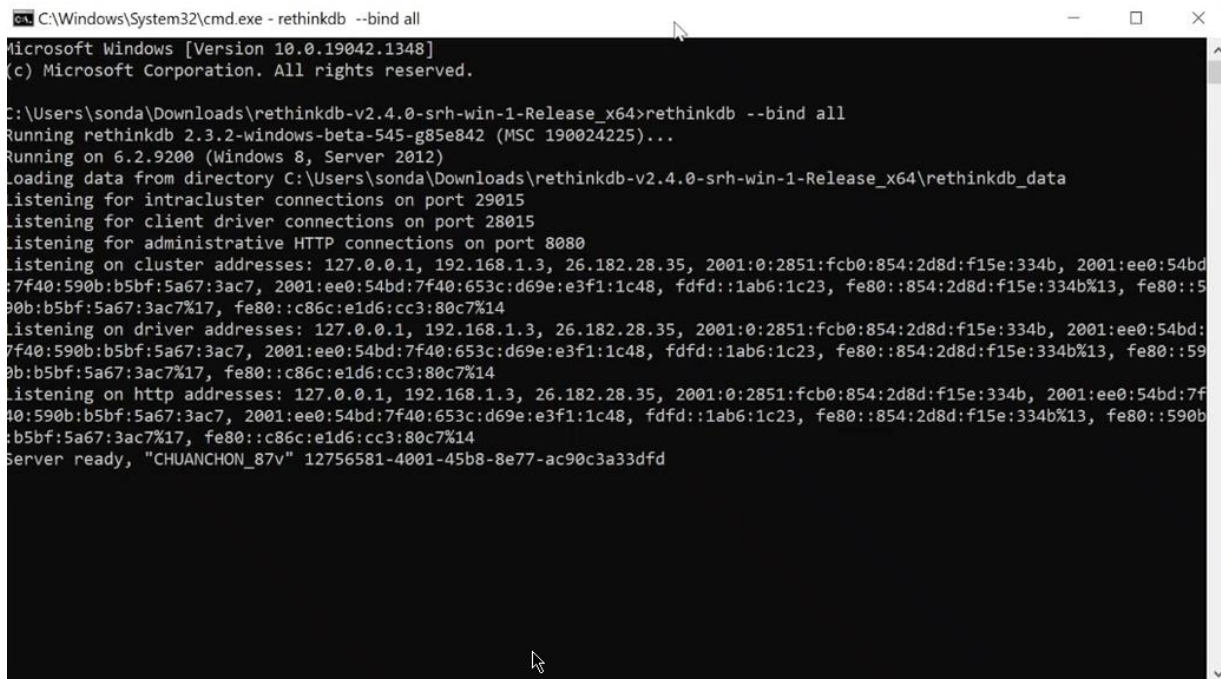
```
r.db('COVID19_2').table('BenhNhan').filter(r.row('MaBN').eq("BN15")).delete()
```

6. Xóa những bệnh nhân được điều trị bởi nhân viên có mã NV14.

```
r.db('COVID19_2').table('BenhNhan').filter(r.row("NhanVienDieuTri").contains('NV14')).delete()
```

2. Kết quả

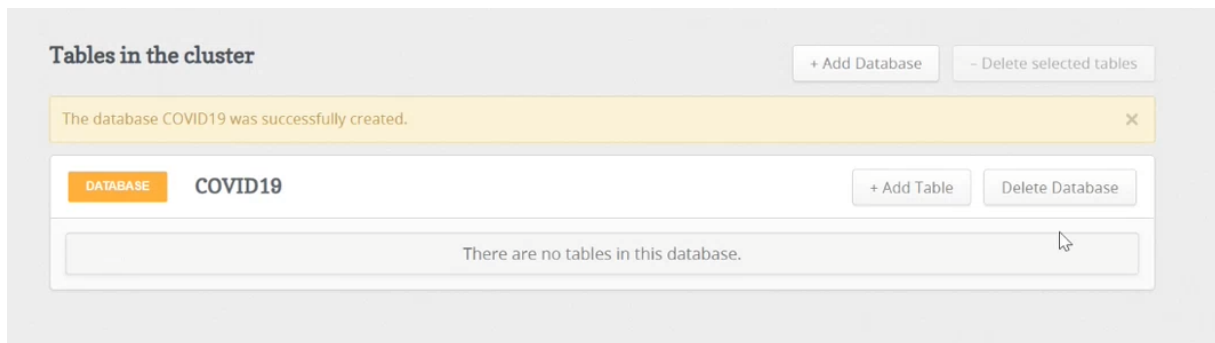
Máy 1 thực hiện lệnh mở server để các máy khác có thể truy cập vào



```
C:\Windows\System32\cmd.exe - rethinkdb --bind all
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

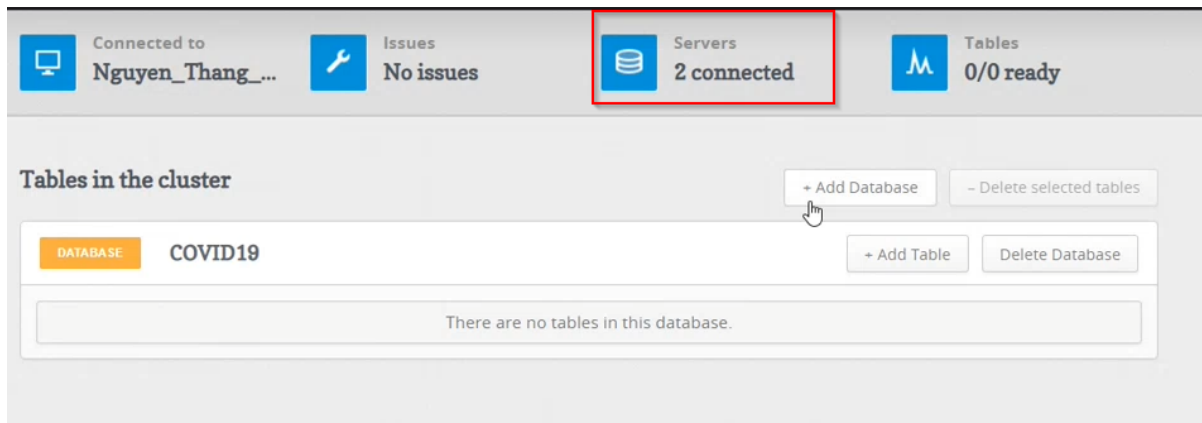
C:\Users\sonda\Downloads\rethinkdb-v2.4.0-srh-win-1-Release_x64>rethinkdb --bind all
Running rethinkdb 2.3.2-windows-beta-545-g85e842 (MSC 190024225)...
Running on 6.2.9200 (Windows 8, Server 2012)
Loading data from directory C:\Users\sonda\Downloads\rethinkdb-v2.4.0-srh-win-1-Release_x64\rethinkdb_data
listening for intracluster connections on port 29015
listening for client driver connections on port 28015
listening for administrative HTTP connections on port 8080
listening on cluster addresses: 127.0.0.1, 192.168.1.3, 26.182.28.35, 2001:0:2851:fc0:854:2d8d:f15e:334b, 2001:ee0:54bd:7f40:590b:b5bf:5a67:3ac7, 2001:ee0:54bd:7f40:653c:d69e:e3f1:1c48, fdfd::1ab6:1c23, fe80::854:2d8d:f15e:334b%13, fe80::590b:b5bf:5a67:3ac7%17, fe80::c86c:e1d6:cc3:80c7%14
listening on driver addresses: 127.0.0.1, 192.168.1.3, 26.182.28.35, 2001:0:2851:fc0:854:2d8d:f15e:334b, 2001:ee0:54bd:7f40:590b:b5bf:5a67:3ac7, 2001:ee0:54bd:7f40:653c:d69e:e3f1:1c48, fdfd::1ab6:1c23, fe80::854:2d8d:f15e:334b%13, fe80::590b:b5bf:5a67:3ac7%17, fe80::c86c:e1d6:cc3:80c7%14
listening on http addresses: 127.0.0.1, 192.168.1.3, 26.182.28.35, 2001:0:2851:fc0:854:2d8d:f15e:334b, 2001:ee0:54bd:7f40:590b:b5bf:5a67:3ac7, 2001:ee0:54bd:7f40:653c:d69e:e3f1:1c48, fdfd::1ab6:1c23, fe80::854:2d8d:f15e:334b%13, fe80::590b:b5bf:5a67:3ac7%17, fe80::c86c:e1d6:cc3:80c7%14
Server ready, "CHUANCHON_87v" 12756581-4001-45b8-8e77-ac90c3a33dfd
```

Máy 1 thực hiện tạo database

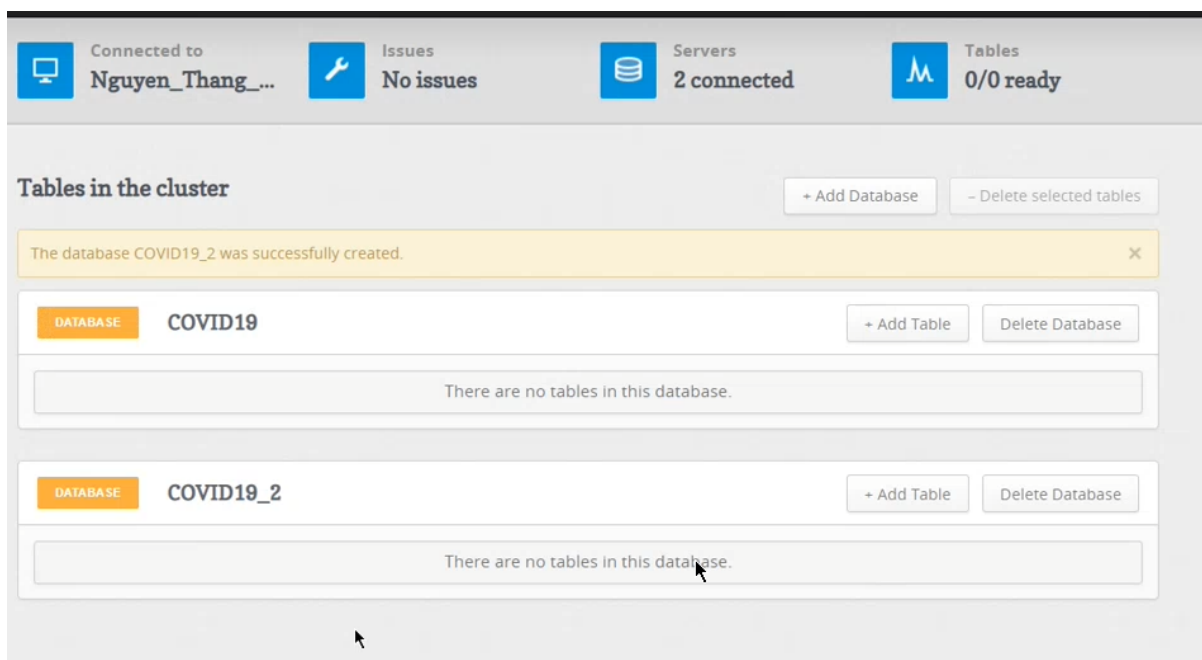


Máy 2 thực hiện kết nối vào máy 1

```
Administrator: Windows PowerShell
Thang at E:\rethinkdb-v2.4.0-srh-win-1-Release_x64_2 .\rethinkdb.exe --join 26.182.28.35 --bind
Running rethinkdb 2.3.2-windows-beta-545-g85e842 (MSC 190024225)...
Running on 6.2.9200 (Windows 8, Server 2012)
Loading data from directory E:\rethinkdb-v2.4.0-srh-win-1-Release_x64_2\rethinkdb_data
Listening for intracluster connections on port 29015
Connected to server "CHUANCHON_87v" 12756581-4001-45b8-8e77-ac90c3a33dfd
Listening for client driver connections on port 28015
Listening for administrative HTTP connections on port 8080
Listening on cluster addresses: 127.0.0.1, 192.168.56.1, 192.168.1.7, 26.156.119.135, 2001:0:2851:fc
b0:34d6:3b31:5407:dd43, 2402:800:63b7:ae55:5fd:235f:91f6:551a, 2402:800:63b7:ae55:209e:8162:b330:591
9, fdfd::1a9c:7787, fe80::1dda:d48b:bbd9:d621%15, fe80::209e:8162:b330:5919%10, fe80::34d6:3b31:5407
:dd43%14, fe80::4472:2bd6:eca4:1574%19
Listening on driver addresses: 127.0.0.1, 192.168.56.1, 192.168.1.7, 26.156.119.135, 2001:0:2851:fc
b0:34d6:3b31:5407:dd43, 2402:800:63b7:ae55:5fd:235f:91f6:551a, 2402:800:63b7:ae55:209e:8162:b330:5919
, fdfd::1a9c:7787, fe80::1dda:d48b:bbd9:d621%15, fe80::209e:8162:b330:5919%10, fe80::34d6:3b31:5407
:dd43%14, fe80::4472:2bd6:eca4:1574%19
Listening on http addresses: 127.0.0.1, 192.168.56.1, 192.168.1.7, 26.156.119.135, 2001:0:2851:fc
b0:34d6:3b31:5407:dd43, 2402:800:63b7:ae55:5fd:235f:91f6:551a, 2402:800:63b7:ae55:209e:8162:b330:5919
, fdfd::1a9c:7787, fe80::1dda:d48b:bbd9:d621%15, fe80::209e:8162:b330:5919%10, fe80::34d6:3b31:5407
:dd43%14, fe80::4472:2bd6:eca4:1574%19
Server ready, "Nguyen_Thang_739" 499c13b8-4da1-4a2c-9360-cb8c4a293ab6
```



Máy 2 thực hiện tạo database



Máy 1 thực hiện tạo bảng và thêm dữ liệu

Tạo bảng



1 row returned in 1.94s.

Tree view Table Viewer View Table view Raw view Query profile

```

{
  "config_changes": [
    {
      "new_val": {
        "db": "COVID19",
        "durability": "hard",
        "id": "3a2e8cb3-5cb6-465a-a0bd-b48d57d57fa1",
        "indexes": [ ],
        "name": "BenhNhan",
        "primary_key": "MaBN",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "Nguyen_Thang_739",
            "replicas": [
              "Nguyen_Thang_739"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      },
      "old_val": null
    }
  ],
  "tables_created": 1
}

```

Data Explorer

History Settings Full Screen

```

1 r.db('COVID19').tableCreate('DiemCachLy', {primaryKey: "MaDCL"})

```

Clear Run

1 row returned in 1.79s.

Tree view Table Viewer View Table view Raw view Query profile

```

{
  "config_changes": [
    {
      "new_val": {
        "db": "COVID19",
        "durability": "hard",
        "id": "2dfc83c4-586b-4187-a85b-07925af4909a",
        "indexes": [ ],
        "name": "DiemCachLy",
        "primary_key": "MaDCL",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "CHUANCHON_87v",
            "replicas": [
              "CHUANCHON_87v"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      },
      "old_val": null
    }
  ]
}

```

Data Explorer History Settings Fullscreen

```
1 r.db('COVID19').tableCreate('NhanVienDieuTri', {primaryKey: "MaNV"})
```

Clear Run

1 row returned in 1.92s.

Tree view Table Viewer View Table view Raw view Query profile

```
{
  "config_changes": [
    {
      "new_val": {
        "db": "COVID19",
        "durability": "hard",
        "id": "23d01e5d-77a8-459d-aa5e-30559c5a50d2",
        "indexes": [ ],
        "name": "NhanVienDieuTri",
        "primary_key": "MaNV",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "Nguyen_Thang_739",
            "replicas": [
              "Nguyen_Thang_739"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      },
      "old_val": null
    }
  ]
}
```

Thêm dữ liệu

Data Explorer History Settings Fullscreen

```
1 r.db('COVID19').table("DiemCachLy").insert(
2   [
3     {
4       "MaDCL": "CL01",
5       "TenDiemCachLy": "Điểm cách ly TP. Dĩ An",
6       "DiaChi": "Dĩ An, Đồng Hòa, Bình Dương"
7     },
8     {
9       "MaDCL": "CL02",
10      "TenDiemCachLy": "Điểm cách ly TP. Biên Hòa",
11      "DiaChi": "30 Nguyễn Trãi, TP. Biên Hòa - Đồng Nai"
12    }
13  ])
```

Clear Run

1 row returned in 60ms.

Tree view Table Viewer View Table view Raw view Query profile

```
{
  "deleted": 0,
  "errors": 0,
  "inserted": 2,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}
```

```

30     "GioiTinh": "Nữ",
31     "DonVi": "CL01"
32   },
33   {
34     "MaNV": "NV06",
35     "HoTen": "Nguyễn Lê Thảo Duyên",
36     "GioiTinh": "Nữ",
37     "DonVi": "CL02"
38   },
39   {
40     "MaNV": "NV07",
41     "HoTen": "Lê Quốc Tín",
42     "GioiTinh": "NNam",
43     "DonVi": "CL02"
44   },
45   {
46     "MaNV": "NV08",
47     "HoTen": "Phạm Hoàng Mỹ",
48     "GioiTinh": "Nữ",
49     "DonVi": "CL01"
50   },
51   {
52     "MaNV": "NV09",
53     "HoTen": "Trần Nhật Tân",
54     "GioiTinh": "Nam",
55     "DonVi": "CL01"
56   },
57   {
58     "MaNV": "NV10",
59     "HoTen": "Phạm Thu Giang",
60     "GioiTinh": "Nữ",
61     "DonVi": "CL02"
62   }
63 ];
64

```

Clear

Run

1 row returned in 164ms.

Tree view

Table Viewer View

Table view

Raw view

Query profile

```

{
  "deleted": 0,
  "errors": 0,
  "inserted": 10,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}

```



```

80  {
81    "MaBN": "BN08",
82    "HoTen": "Phan Văn Tài",
83    "TrieuChung": [
84      "Ho", "Sốt"
85    ],
86    "GioiTinh": "Nam",
87    "NuocVe": "Trung Quốc",
88    "DiemCachLy": "CL01",
89    "NhanVienDieuTri": ["NV05", "NV07"]
90  },
91  {
92    "MaBN": "BN09",
93    "HoTen": "Nguyễn Minh An",
94    "TrieuChung": [
95      "Sốt"
96    ],
97    "GioiTinh": "Nam",
98    "NuocVe": "Mỹ",
99    "DiemCachLy": "CL02",
100   "NhanVienDieuTri": ["NV06", "NV08"]
101  },
102  {
103    "MaBN": "BN10",
104    "HoTen": "Nguyễn Tấn Khang",
105    "TrieuChung": [
106      "Tức ngực", "Khó thở", "Ho", "Sốt"
107    ],
108    "GioiTinh": "Nam",
109    "NuocVe": "Anh",
110    "DiemCachLy": "CL01",
111    "NhanVienDieuTri": ["NV09", "NV10"]
112  }
113  ]});
114

```

Clear

Run

1 row returned in 149ms.

Tree view

Table Viewer View

Table view

Raw view

Query profile

```

{
  "deleted": 0 ,
  "errors": 0 ,
  "inserted": 10 ,
  "replaced": 0 ,
  "skipped": 0 ,
  "unchanged": 0
}

```

Máy 2 thực hiện tạo bảng và thêm dữ liệu

Tạo bảng

```
1 r.db('COVID19_2').tableCreate('BenhNhan', {primaryKey: "MaBN"}) ;
2
3 r.db('COVID19_2').tableCreate('DiemCachLy', {primaryKey: "MaDCL"}) ;
4
5 r.db('COVID19_2').tableCreate('NhanVienDieuTri', {primaryKey: "MaNV"}) ;
```

Clear Run

1 row returned in 1.96s.

Tree view Table Viewer View Table view Raw view Query profile

```
{
  "config_changes": [
    {
      "new_val": {
        "db": "COVID19_2",
        "durability": "hard",
        "id": "4d7e5e64-5e50-4384-8649-f57851b8c3a7",
        "indexes": [ ],
        "name": "NhanVienDieuTri",
        "primary_key": "MaNV",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "CHUANCHON_87v",
            "replicas": [
              "CHUANCHON_87v"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      },
      "old_val": null
    }
  ],
  "tables_created": 1
}
```

Data Explorer

History Settings

```
1 r.db('COVID19_2').table("DiemCachLy").insert(
2
3 [
4
5 {
6
7   "MaDCL": "CL03",
8
9   "TenDiemCachLy": "Điểm cách ly TP.Thủ Đức",
10
11   "DiaChi": "30 Võ Văn Ngân, TP.Thủ Đức"
12
13 },
14
15 {
16
17   "MaDCL": "CL04",
18
19   "TenDiemCachLy": "Điểm cách ly huyện Củ Chi",
20
21   "DiaChi": "ấp 4, huyện Củ Chi, TP HCM"
22
23 }
24 ]
25
26
27 );
```

Clear Run

1 row returned in 59ms.

Tree view Table Viewer View Table view Raw view Query profile

```
{
  "deleted": 0,
  "errors": 0,
  "inserted": 2,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}
```

```
195     "MaDCL": "CL04",
196
197     "NhanVienDieuTri": ["NV17"],
198
199   },
200
201   {
202     "MaBN": "BN20",
203     "HoTen": "Ngô Bá Khả",
204     "TrieuChung": [
205       "Ho", "Tức ngực", "Khó thở"
206     ],
207     "NucVe": "Tây Ban Nha",
208     "GioiTinh": "Nam",
209     "MaDCL": "CL04",
210     "NhanVienDieuTri": ["NV19", "NV20"],
211   },
212
213 ]}
214
215
216
217
218
219
220
221
222
223
224
225
```

Clear Run

1 row returned in 145ms.

Tree view Table Viewer View Table view Raw view Query profile

```
{
  "deleted": 0,
  "errors": 0,
  "inserted": 10,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}
```

```
99 {
100
101   "MaNV": "NV19",
102   "HoTen": "Ngô La Thiện Trí",
103   "DonVi": "CL04",
104   "GioiTinh": "Nam"
105 },
106
107 {
108   "MaNV": "NV20",
109   "HoTen": "Hà Hồ Thiên Giới",
110   "DonVi": "CL04",
111   "GioiTinh": "Nữ"
112 },
113
114 ];
115
116
117
118
119
120
121
122
123
124
125
```

Clear Run

1 row returned in 135ms.

Tree view Table Viewer View Table view Raw view Query profile

```
{
  "deleted": 0,
  "errors": 0,
  "inserted": 10,
  "replaced": 0,
  "skipped": 0,
  "unchanged": 0
}
```

Máy 1 thực hiện các câu select, update, delete

Select

Cho biết những bệnh nhân có triệu chứng ho và sốt.

Data Explorer History Settings

```
1 r.db('COVID19_2').table('BenhNhan').filter(r.row('TrieuChung').contains('Ho','Sot'))
```

Clear Run

4 rows returned. Displaying rows 0-3

Tree view Table Viewer View Table view Raw view Query profile

	GioiTinh	HoTen	MaBN	MaDCL	NhanVienDieuTri	NuocVe	TrieuChung
0	Nữ	Nguyễn Phan Gia Hân	BN13	CL03	[...]	Trung Quốc	[...]
1	Nam	Nguyễn Văn Hoài Tử	BN18	CL04	[...]	Canada	[...]
2	Nam	Võ Nguyễn Hoài Linh	BN16	CL04	[...]	Trung Quốc	[...]
3	Nữ	Lê Kim Duyên	BN14	CL03	[...]	Anh	[...]

Cho biết những bệnh nhân có nhiều hơn 3 triệu chứng và đang điều trị tại TP. Thủ Đức.

Data Explorer History Settings

```
1 r.db('COVID19_2').table("BenhNhan").eqJoin("MaDCL", r.db('COVID19_2').table("DiemCachLy")).filter(function(benhnhnan){
2 return benhnhnan('left')['TrieuChung'].count().gt(3).and(benhnhnan('right')['TenDiemCachLy'].eq('Điểm cách ly TP.Thủ Đức
3 c'))
4 })
```

Clear Run

1 row returned.

Tree view Table Viewer View Table view Raw view Query profile

	left.GioiTinh	left.HoTen	left.MaBN	left.MaDCL	left.NhanVienDieuTri	left.NuocVe	left.TrieuChung
0	Nữ	Nguyễn Phan Gia Hân	BN13	CL03	[...]	Trung Quốc	[...]

Update

Cập nhật triệu chứng Mất vị giác, Ho, Tức ngực cho bệnh nhân có mã là BN13.

Data Explorer History Settings

```
1 r.db('COVID19_2').table('BenhNhan').filter(r.row('MaBN').eq("BN13")).update({"TrieuChung":["Mất vị giác", "Ho", "Tức Ngực"]})
```

Clear Run

1 row returned in 41ms. Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	0	0	0	1	0	0

Cập nhật bệnh nhân có mã là BN15 được điều trị bởi nhân viên có mã là NV11.

Data Explorer History Settings

```
1 r.db('COVID19_2').table('BenhNhan').filter(r.row('MaBN').eq("BN15")).update({"NhanVienDieuTri":["NV11"]})
```

Clear Run

1 row returned in 41ms. Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	0	0	0	1	0	0

Delete

Xóa bệnh nhân có mã là BN15

Data Explorer History Settings

```
1 r.db('COVID19_2').table('BenhNhan').filter(r.row('MaBN').eq("BN15")).delete()
```

Clear Run

1 row returned in 41ms. Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	1	0	0	0	0	0

Xóa những bệnh nhân được điều trị bởi nhân viên có mã NV14.

Data Explorer History Settings Fullscreen

```
1 r.db('COVID19_2').table('BenhNhan').filter(r.row("NhanVienDieuTri").contains('NV14')).delete()
```

Clear Run

1 row returned in 40ms. Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	1	0	0	0	0	0

Máy 2 thực hiện các câu select, update, delete

Select

Cho biết những bệnh nhân có triệu chứng ho và sốt.

Data Explorer History Settings Fullscreen

```
1 r.db('COVID19').table('BenhNhan').filter(r.row('TrieuChung').contains('Ho','Sốt'))
```

Clear Run

Tree view Table Viewer View Table view Raw view Query profile

	DiemCachLy	GioiTinh	HoTen	MaBN	NhanVienDieu...	NuocVe	TrieuChung
0	CL02	Nữ	Lê Kim Duyên	BN02	> [...]	Hàn Quốc	> [...]
1	CL01	Nam	Phan Văn Tài	BN08	> [...]	Trung Quốc	> [...]
2	CL01	Nam	Nguyễn Văn H...	BN01	> [...]	Trung Quốc	> [...]
3	CL01	Nam	Đỗ Văn Nam	BN05	> [...]	Canada	> [...]
4	CL01	Nam	Nguyễn Tấn K...	BN10	> [...]	Anh	> [...]

End of results

Cho biết những bệnh nhân có nhiều hơn 3 triệu chứng và đang điều trị tại Dĩ An.

Data Explorer Clear history History Settings

```
{
  "MaNV": "NV19",
  "HoTen": "Ngô La Thiện Trí",
  "DonVi": "CLO4",
  "GioiTinh": "Nam"
},
{
  "MaNV": "NV20",
  "HoTen": "Hạ Hồ Thiên Giới",
  "DonVi": "CLO4",
  "GioiTinh": "Nữ"
},
}
```

```
1 r.db('COVID19').table('BenhNhan').eqJoin("MaDCL", r.db('COVID19').table("DiemCachLy")).filter(function(benhnhnhan){
2
3 return benhnhnhan('left')('TrieuChung').count().gt(3).and(benhnhnhan('right')('TenDiemCachLy').eq('Điểm cách ly TP. Dĩ An'))
4
5 }}
```

Clear Run

2 rows returned. Displaying rows 0-1

Tree view Table Viewer View Table view Raw view Query profile

	left.GioiTinh	left.HoTen	left.MaBN	left.MaDCL	left.NhanVienDieuTri	left.NuocVe	left.TrieuChung	right.TenDiemCachLy
0	Nam	Đỗ Văn Nam	BN05	CL01	[...]	Canada	[...]	Điểm cách ly TP. Dĩ An
1	Nam	Nguyễn Tấn Khang	BN10	CL01	[...]	Anh	[...]	Điểm cách ly TP. Dĩ An

Update

Cập nhật triệu chứng Mất vị giác, Ho, Tức ngực cho bệnh nhân có mã là BN06.

Data Explorer Clear history History Settings

```
{
  "MaNV": "NV19",
  "HoTen": "Ngô La Thiện Trí",
  "DonVi": "CLO4",
  "GioiTinh": "Nam"
},
{
  "MaNV": "NV20",
  "HoTen": "Hạ Hồ Thiên Giới",
  "DonVi": "CLO4",
  "GioiTinh": "Nữ"
},
}
```

```
1
2
3 r.db('COVID19').table('BenhNhan').filter(r.row('MaBN').eq("BN06")).update({"TrieuChung":["Mất vị giác", "Ho", "Tức Ngực"]})
4
5
```

Clear Run

1 row returned in 53ms.

Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	0	0	0	1	0	0

Cập nhật bệnh nhân có mã là BN05 được điều trị bởi nhân viên có mã là NV02.

Data Explorer Clear history History Settings Fullscreen

```
{
  "MaNV": "NV19",
  "HoTen": "Ngô La Thiện Trí",
  "DonVi": "CLO4",
  "GioiTinh": "Nam"
},
{
  "MaNV": "NV20",
  "HoTen": "Hà Hồ Thiên Giới",
  "DonVi": "CLO4",
  "GioiTinh": "Nữ"
},
}
```

```
1 r.db('COVID19').table('BenhNhan').filter(r.row('MaBN').eq('BN05')).update({'NhanVienDieuTri':['NV02']})
```

Clear Run

1 row returned in 58ms.

Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	0	0	0	1	0	0

Delete

Xóa bệnh nhân có mã là BN03

Data Explorer Clear history History Settings Fullscreen

```
{
  "MaNV": "NV19",
  "HoTen": "Ngô La Thiện Trí",
  "DonVi": "CLO4",
  "GioiTinh": "Nam"
},
{
  "MaNV": "NV20",
  "HoTen": "Hà Hồ Thiên Giới",
  "DonVi": "CLO4",
  "GioiTinh": "Nữ"
},
}
```

```
1 r.db('COVID19').table('BenhNhan').filter(r.row('MaBN').eq('BN03')).delete()
```

Clear Run

1 row returned in 53ms.

Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	1	0	0	0	0	0

Xóa những bệnh nhân được điều trị bởi nhân viên có mã NV01.

Data Explorer

Clear history

History

```

{
  "MaNV": "NV19",
  "HoTen": "Ngô La Thiện Trí",
  "DonVi": "CL04",
  "GioiTinh": "Nam"
},
{
  "MaNV": "NV20",
  "HoTen": "Hà Hồ Thiên Giới",
  "DonVi": "CL04",
  "GioiTinh": "Nữ"
}

```

Remove

Load

```

1
2
3
r.db('COVID19').table('BenhNhan').filter(r.row("NhanVienDieuTri").contains('NV01')).delete()

```

Clear

Run

1 row returned in 246ms.

Tree view

Table Viewer View

Table view

Raw view

Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	2	0	0	0	0	0

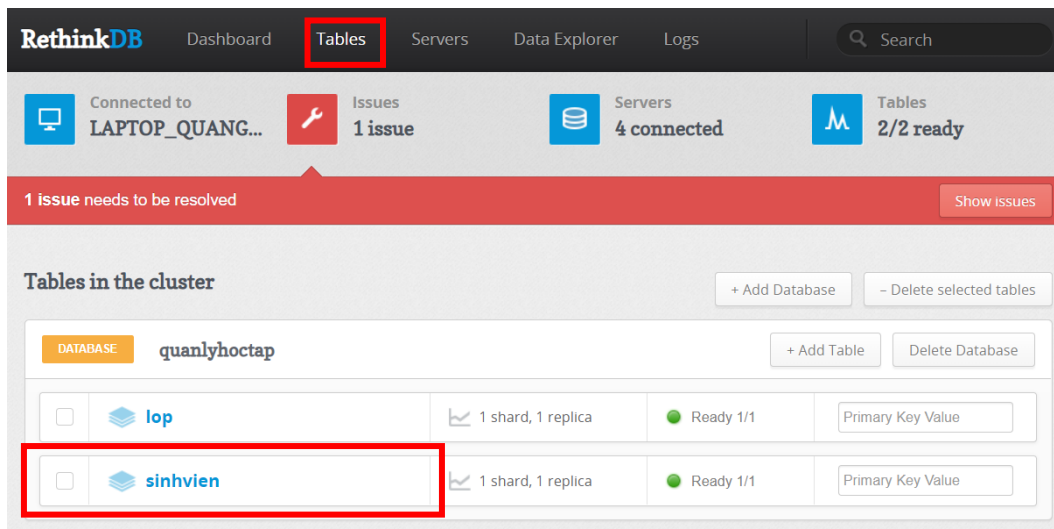
3. Cơ chế nhân bản trong phân tán RethinkDB

Nhắc lại kiến thức về nhân bản. Nhân bản là việc thực hiện copy dữ liệu từ server này sang server khác. Mục đích của việc nhân bản là để dự phòng và tăng tính khả dụng của dữ liệu

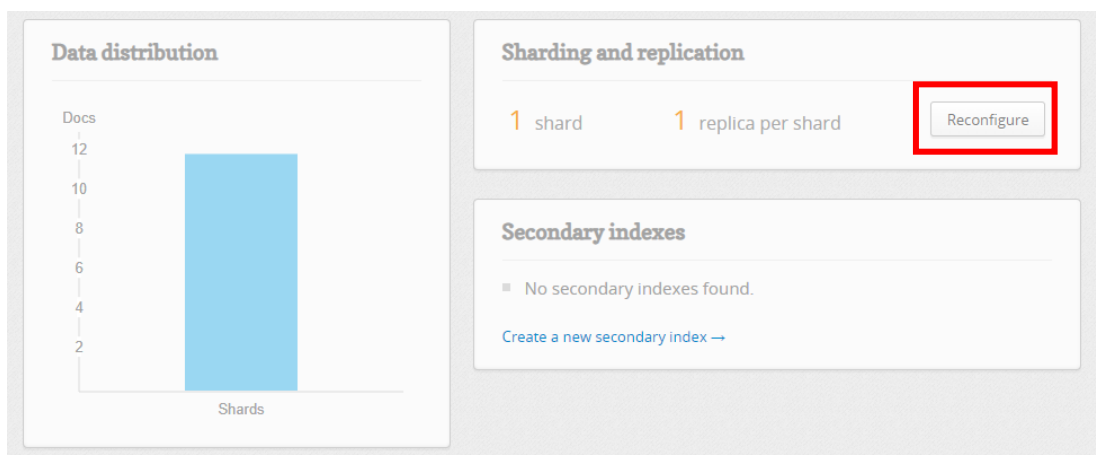
Trong RethinkDB, việc nhân bản chỉ ở 'table'. Khi dữ liệu nhân bản được lưu trên nhiều server khác nhau thì sẽ có một server chính (primary). Ví dụ như có một nhân bản được lưu ở server chính và server phụ. Khi một trong hai server gặp sự cố thì chúng ta vẫn có thể truy cập dữ liệu nhân bản ở server còn lại. Ngoài ra, RethinkDB tự động cân bằng các câu truy vấn giữa các nhân bản, giảm lưu lượng truy cập trên mỗi server và tăng hiệu suất tổng thể.

Các bước để nhân bản một tables bằng cách sử dụng giao diện web của RethinkDB:

Bước 1: Chọn Tables và chọn bảng cần muốn nhân bản



Bước 2: Nhấn vào nút Reconfigure.



Bước 3: Chọn số lượng bản sao cần nhân bản. Trước khi cập nhật cấu hình, hãy nhấp vào mục *Where's my data going?* để xem dữ liệu sẽ được lan truyền như thế nào. Chẳng hạn như ví dụ này hiển thị 1 phân đoạn với 4 bản sao:

Sharding and replication for quanlyhoctap.sinhvien

Applying changes may cause the table to become temporarily unavailable.

1shards

4replicas per shard

Where's my data going?

Shard 1

- Nguyen_Thang_6a7
- CHUANCHON_ue8
- HongCuc_Oct
- LAPTOP_QUANGHUY

Primary replica

Secondary replica

Secondary replica

Secondary replica

Cancel

Apply configuration

Lưu ý cách các bản sao được trải ra ở nhiều nơi khác nhau - nếu có điều gì đó xảy ra trên một nút, các bản sao khác của chúng ta vẫn an toàn và bình thường.

Bước 4: Nhấn nút Apply configuration để áp dụng cấu hình.

TÀI LIỆU THAM KHẢO

- [1] "A Practical Introduction to RethinkDB | Pluralsight | Pluralsight."
<https://www.pluralsight.com/guides/a-practical-introduction-to-rethinkdb>
(accessed Dec. 26, 2021).
- [2] "SQL to ReQL cheat sheet - RethinkDB." <https://rethinkdb.com/docs/sql-to-reql/python/> (accessed Dec. 26, 2021).
- [3] "Introduction to ReQL - RethinkDB." <https://rethinkdb.com/docs/introduction-to-reql/> (accessed Dec. 27, 2021).
- [4] "Architecture FAQ - RethinkDB." <https://rethinkdb.com/docs/architecture/> (accessed Dec. 27, 2021).
- [5] "database replication - Connect to and work with a RethinkDB cluster," *Stack Overflow*. <https://stackoverflow.com/questions/29741326/connect-to-and-work-with-a-rethinkdb-cluster> (accessed Dec. 27, 2021).
- [6] G. Tiepolo, *Getting Started with RethinkDB*. Packt Publishing Ltd, 2016.
- [7] "RethinkDB," *Wikipedia*. Dec. 02, 2021. Accessed: Dec. 26, 2021. [Online]. Available:
<https://en.wikipedia.org/w/index.php?title=RethinkDB&oldid=1058283319>
- [8] Traversy Media, *RethinkDB & ReQL Crash Course*, (2019). Accessed: Dec. 27, 2021. [Online Video]. Available:
<https://www.youtube.com/watch?v=pW3PFtchHDc>
- [9] "Scaling, sharding and replication - RethinkDB."
<https://rethinkdb.com/docs/sharding-and-replication/> (accessed Dec. 27, 2021).

- [10] "NoSQL là gì? | Cơ sở dữ liệu phi quan hệ, Mô hình dữ liệu sơ đồ linh hoạt | AWS," *Amazon Web Services, Inc.* <https://aws.amazon.com/vi/nosql/> (accessed Dec. 27, 2021).
- [11] "Giới thiệu về hệ quản trị cơ sở dữ liệu RethinkDB," *Viblo*, May 14, 2018. <https://viblo.asia/p/gioi-thieu-ve-he-quan-tri-co-so-du-lieu-rethinkdb-jvEla4O6Zkw> (accessed Dec. 26, 2021).
- [12] RethinkDB, *RethinkDB dành cho Windows: bản xem trước dành cho nhà phát triển*, (2016). Accessed: Dec. 27, 2021. [Online Video]. Available: <https://www.youtube.com/watch?v=klgE2Jo5beE>