

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



BÀI TẬP LỚN 2
MÔN HỌC: CƠ SỞ DỮ LIỆU PHÂN TÁN
ĐỀ TÀI:
TÌM HIỂU HỆ QUẢN TRỊ CSDL NOSQL – RETHINKDB

Giảng viên hướng dẫn: **Lê Đức Tín**

Lớp: **IS211.011**

Nhóm sinh viên thực hiện: **Nguyễn Anh Phi - 19522005**

Nguyễn Thy Anh Tài - 20521860

Nguyễn Phước Thắng - 21522590

Nguyễn Quang Huy – 20521403

Thành phố Hồ Chí Minh, tháng 1 năm 2024

LỜI CẢM ƠN

Lời đầu tiên, nhóm em chân thành cảm ơn anh Lê Đức Tín 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 Tín 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 kỹ 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 anh, 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 để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Nhóm thực hiện,

Anh Phi – Anh Tài - Phước Thắng – Quang Huy.

MỤC LỤC

LỜI CẢM ƠN	1
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	3
1. GIỚI THIỆU NOSQL	3
2. GIỚI THIỆU CSDL RETHINKDB	6
2.1 LỊCH SỬ RA ĐỜI	6
2.2 TỔNG QUAN – ƯU NHƯỢC ĐIỂM	6
2.3 SO SÁNH RETHINKDB VỚI CÁC HQT CSDL KHÁC	7
2.4. GIỚI THIỆU GIAO DIỆN	8
2.5 CÁC THAO TÁC TRÊN RETHINKDB BẰNG NGÔN NGỮ JAVASCRIPT ..	9
2.6 CƠ CHẾ PHÂN TÁN CỦA RETHINKDB	25
CHƯƠNG 2: CÀI ĐẶT VÀ KẾT NỐI NHIỀU MÁY	26
1. CÀI ĐẶT RETHINKDB TRÊN WINDOWS	26
2. KẾT NỐI CÁC MÁY QUA MẠNG LAN ẢO	27
CHƯƠNG 3: TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN	28
1. NHẬP LIỆU TRÊN TỪNG MÁY	28
1.1 Nhập liệu trên máy 1	28
1.2 Nhập liệu trên máy 2	36
2. TRUY VẤN GIỮA HAI MÁY	44
2.1. Máy 1 thực hiện	44
2.2. Máy 2 thực hiện	46
3. CƠ CHẾ NHÂN BẢN TRONG PHÂN TÁN TRONG RETHINKDB	49
CHƯƠNG 4: TÀI LIỆU THAM KHẢO	52

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1. GIỚI THIỆU NOSQL

1.1. Cơ sở dữ liệu NoSQL

Là CSDL 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 một lớn. Ví dụ như Google, Facebook phải lưu trữ và xử lý một lượng dữ liệu cực lớn mỗi ngày.

1.2. 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 CSDL RETHINKDB

2.1. LỊCH SỬ RA ĐỜI

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 – ƯU NHƯỢC ĐIỂM

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. SO SÁNH RETHINKDB VỚI CÁC HQT CSDL KHÁC

- RethinkDB vs MongoDB:

RethinkDB: RethinkDB là một HQT CSDL phân tán hướng tài liệu (document-oriented) với truy vấn thời gian thực và khả năng đồng bộ. Nó cung cấp một ngôn ngữ truy vấn linh hoạt và hỗ trợ truy vấn phức tạp.

MongoDB: MongoDB cũng là một HQT CSDL hướng tài liệu phân tán. Nó được thiết kế để lưu trữ và truy vấn dữ liệu tài liệu linh hoạt. MongoDB cung cấp khả năng mở rộng tốt và hỗ trợ các tính năng như replica set và sharding.

- RethinkDB vs PostgreSQL:

RethinkDB: RethinkDB là một HQT CSDL phân tán hướng tài liệu. Nó tập trung vào tính linh hoạt, truy vấn thời gian thực và đồng bộ dữ liệu.

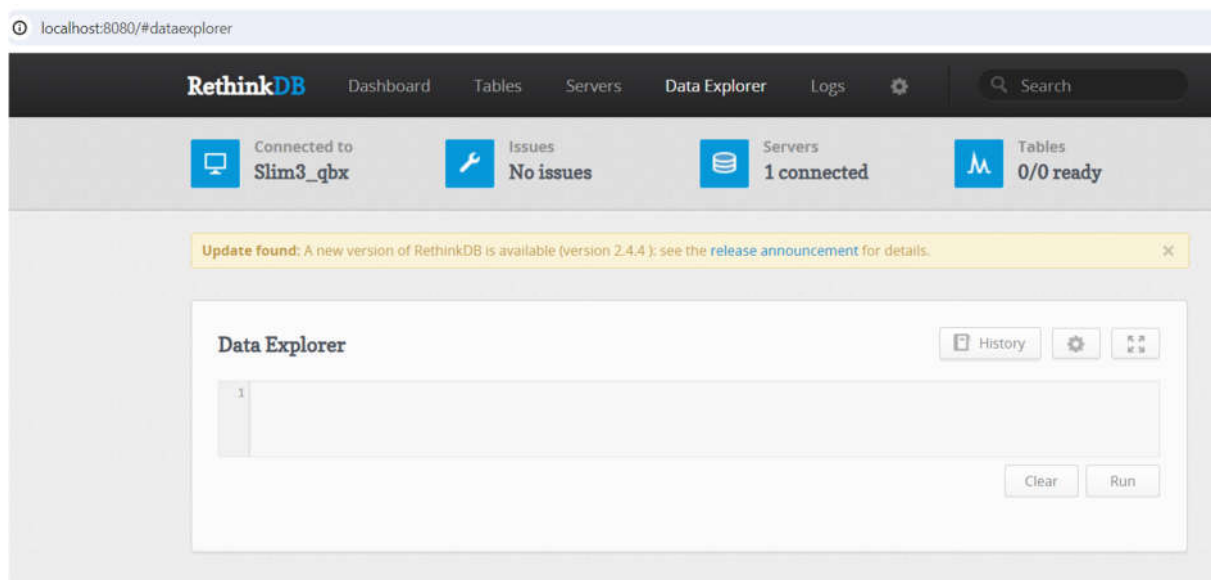
PostgreSQL: PostgreSQL là một HQT CSDL quan hệ (relational database) mạnh mẽ. Nó hỗ trợ các tính năng quan hệ truyền thống như khóa ngoại, giao dịch ACID, và truy vấn SQL phức tạp. PostgreSQL thích hợp cho các ứng dụng yêu cầu tính toàn vẹn dữ liệu và quan hệ phức tạp.

- RethinkDB vs Apache Cassandra:

RethinkDB: RethinkDB là một HQT CSDL phân tán với truy vấn thời gian thực và đồng bộ dữ liệu. Nó tập trung vào tính linh hoạt và dễ sử dụng.

Apache Cassandra: Apache Cassandra là một HQT CSDL phân tán được thiết kế cho khả năng mở rộng và khả năng chịu lỗi. Nó hỗ trợ việc lưu trữ và truy vấn dữ liệu phân tán trên nhiều nút và hướng đến việc cung cấp sự ổn định và khả năng mở rộng.

2.4. GIỚI THIỆU GIAO DIỆN



Giao diện RethinkDB Administration Console với thanh điều hướng gồm có:

- **Dashboard:** chứa thông tin tổng quan của hệ thống, cho biết số lượng server đang kết nối/bị thiếu, số lượng bảng có thể dùng/có vấn đề, số lượng chỉ mục sẵn

sàng/đang thiết kế, mức sử dụng tài nguyên, hiệu suất của cluster, bản log của hệ thống.

- **Tables:** quản lý các database và các bảng trong đó, cung cấp các thao tác liên quan đến bảng (thêm, xóa, sửa, phân mảnh, nhân bản, thêm chỉ mục,...).
- **Servers:** quản lý các server đang kết nối vào cluster, xem thông phân mảnh, nhân bản của mỗi server.
- **Data Explorer:** cung cấp phương tiện truy vấn và thực thi các câu lệnh ReQL.
- **Logs:** ghi nhận lại các bản log, hoạt động của hệ thống.

2.5. CÁC THAO TÁC TRÊN RETHINKDB BẰNG NGÔN NGỮ JAVASCRIPT

2.5.1 Tạo database

- Cách 1: Dùng Data Explorer, ngôn ngữ ReQL

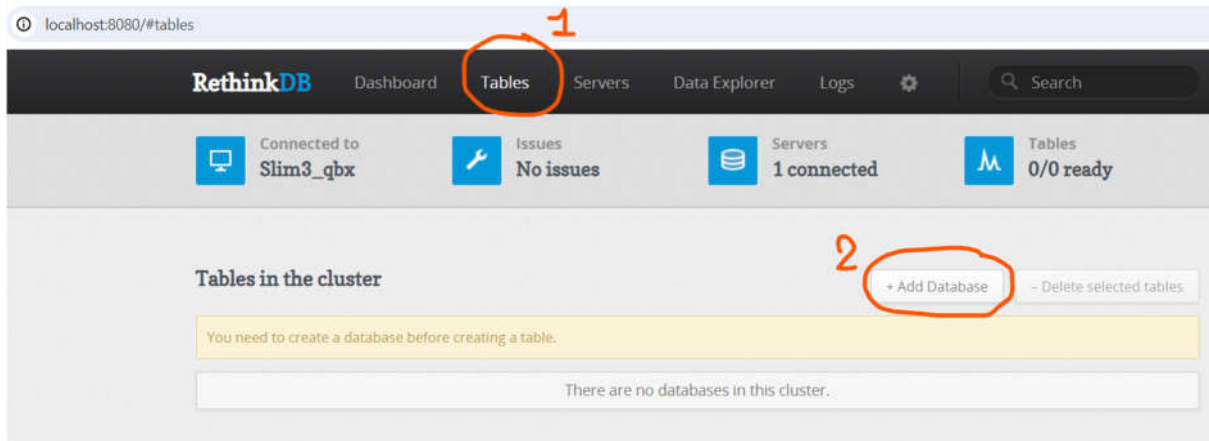
CÚ PHÁP: `r.dbCreate('<TÊN DATABASE>')`

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

The screenshot shows the Data Explorer interface. At the top, there's a header with 'Data Explorer' and buttons for 'History', 'Settings', and 'Full Screen'. Below the header is a text input area containing the command `1 r.dbCreate('JIANGNAM_BAKERY');`. To the right of the input are 'Clear' and 'Run' buttons. Below the input, it says '1 row returned in <1ms.' 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 response:

```
{
  "config_changes": [
    {
      "new_val": {
        "id": "001c6860-490b-40be-b638-0b6bcc38f00b",
        "name": "JIANGNAM_BAKERY"
      },
      "old_val": null
    }
  ],
  "dbs_created": 1
}
```

- Cách 2: Dùng giao diện RethinkDB Administration Console



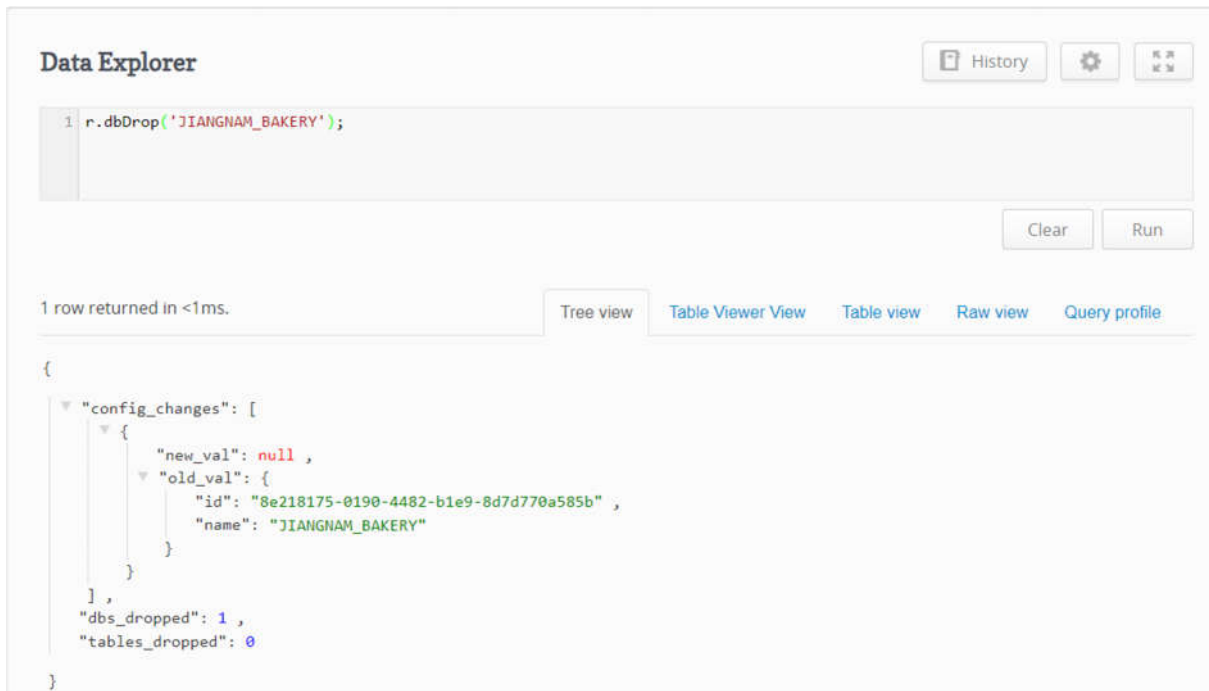
- Bước 1: Từ giao diện chính của RethinkDB ấn vào mục Tables
- Bước 2: Click chọn Add Database
- Bước 3: Nhập tên của Database sắp được tạo, ở đây database có tên là 'JIANGNAM_BAKERY'
- Bước 4: Click Add để tạo database có tên được nhập ở bước 3

2.5.2 Xóa database

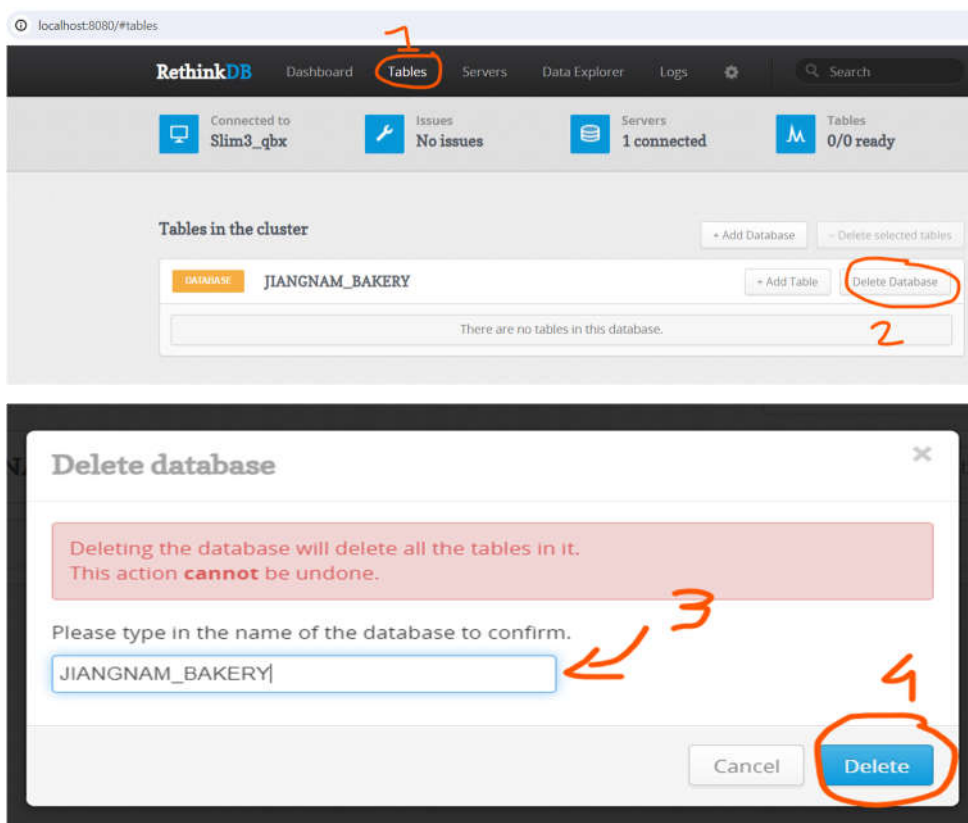
- Cách 1: Dùng Data Explorer, ngôn ngữ ReQL

CÚ PHÁP: `r.dbDrop('<TÊN DATABASE>')`

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



- Cách 2: Dùng giao diện RethinkDB Administration Console



- Bước 1: Từ giao diện chính của RethinkDB ấn vào mục Tables
- Bước 2: Click chọn Delete Database
- Bước 3: Nhập tên của database cần xóa, ở đây cần xóa database có tên là 'JIANGNAM_BAKERY'
- Bước 4: Click chọn delete để xóa database

2.5.3. Tạo bảng

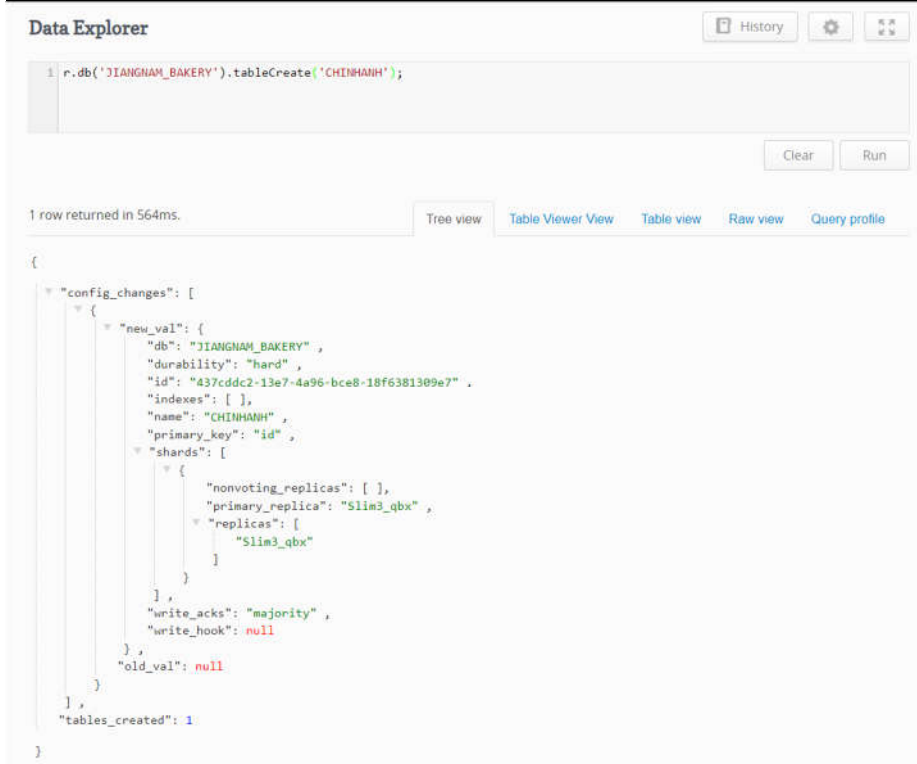
- Cách 1: Dùng Data Explorer, ngôn ngữ ReQL

CÚ PHÁP: `r.db('<TÊN DATABASE>').tableCreate('<TÊN BẢNG>')`

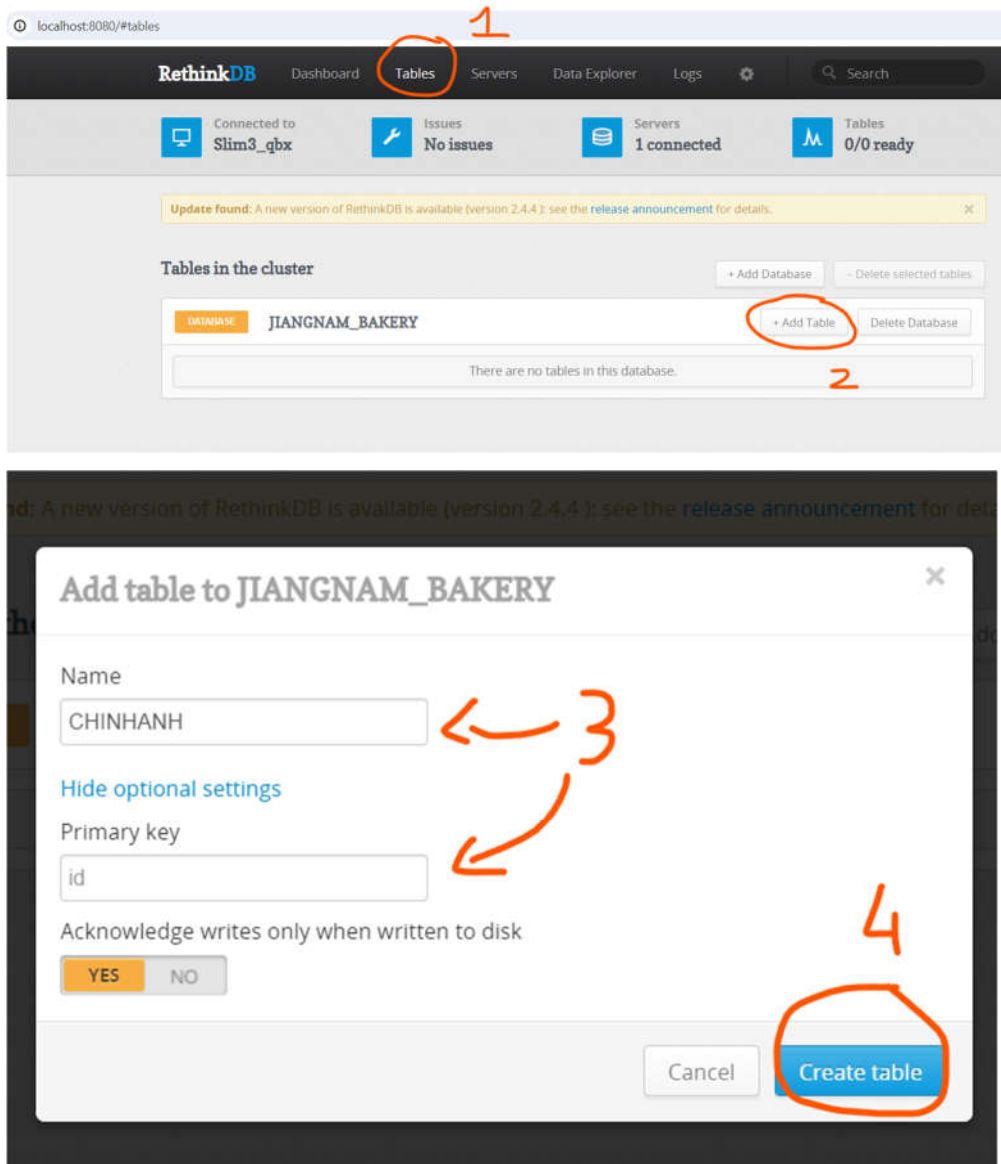
LƯU Ý:

- Nếu bỏ qua `db('<TÊN DATABASE>')`, RethinkDB mặc định sử dụng database 'test' được khởi tạo sẵn.
- RethinkDB là NoSQL database và không dùng lược đồ (schema), do đó không khắt khe về cấu trúc của các document.
- Nếu không khai báo khóa chính, mặc định khóa chính sẽ là "id".

SQL	ReQL
CREATE TABLE CHINHANH(MACN VARCHAR PRIMARY KEY);	<code>r.db('JIANGNAM_BAKERY').tableCreate('CHINHANH', {primaryKey: "MACN"});</code>



- Cách 2: Dùng giao diện RethinkDB Administration Console



- Bước 1: Từ giao diện chính của RethinkDB ấn vào mục Tables
- Bước 2: Chọn database cần được tạo bảng, và click vào Add Table
- Bước 3: Nhập tên bảng ở mục Name và khóa chính ở mục Primary Key
- Bước 4: Ấn vào Create table để tạo bảng

2.5.4. Xóa bảng

- Cách 1: Dùng Data Explorer, ngôn ngữ ReQL

CÚ PHÁP: `r.db('<TÊN DATABASE>').tableDrop('<TÊN BẢNG>')`

LƯU Ý:

- Nếu bỏ qua `db('<TÊN DATABASE>')`, RethinkDB mặc định sử dụng database 'test' được khởi tạo sẵn.

SQL	ReQL
DROP TABLE CHINHANH;	r.db('JIANGNAM_BAKERY').tableDrop('CHINHANH');

Data Explorer History Settings Fullscreen

```
1 r.db('JIANGNAM_BAKERY').tableDrop('CHINHANH');
```

Clear Run

1 row returned in 8ms. Tree view Table Viewer View Table view Raw view Query profile

```
{
  "config_changes": [
    {
      "new_val": null,
      "old_val": {
        "db": "JIANGNAM_BAKERY",
        "durability": "hard",
        "id": "416963d7-f9fb-47af-bc79-b6ca7cc90270",
        "indexes": [ ],
        "name": "CHINHANH",
        "primary_key": "id",
        "shards": [
          {
            "nonvoting_replicas": [ ],
            "primary_replica": "Slim3_qbx",
            "replicas": [
              "Slim3_qbx"
            ]
          }
        ],
        "write_acks": "majority",
        "write_hook": null
      }
    ]
  ],
  "tables_dropped": 1
}
```

- Cách 2: Dùng giao diện RethinkDB Administration Console

localhost:8080/#tables

RethinkDB Dashboard **Tables** Servers Data Explorer Logs Settings Search

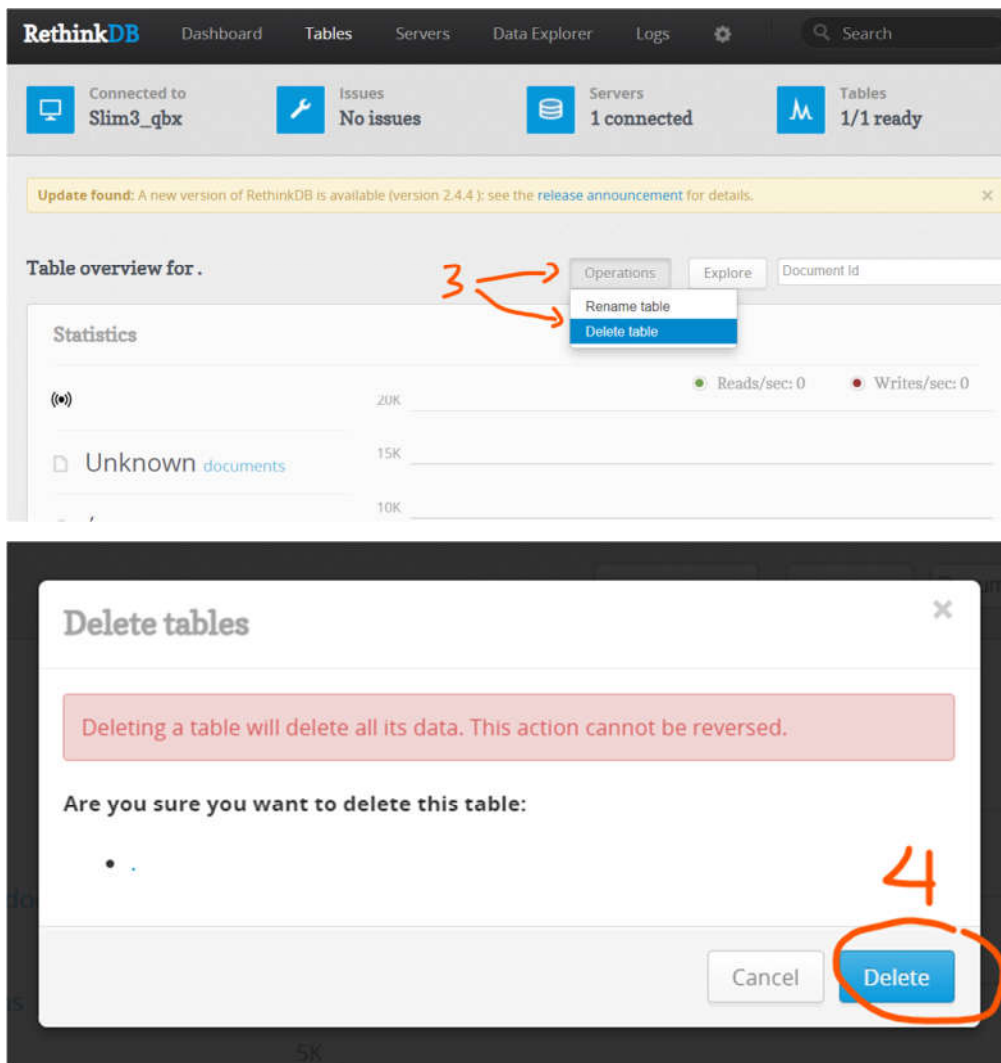
Connected to Slim3_qbx No issues Servers 1 connected Tables 1/1 ready

Update found: A new version of RethinkDB is available (version 2.4.4); see the [release announcement](#) for details.

Tables in the cluster + Add Database - Delete selected tables

DATABASE **JIANGNAM_BAKERY** + Add Table Delete Database

<input type="checkbox"/>	CHINHANH	1 shard, 1 replica	Ready 1/1	Primary Key Value
--------------------------	-----------------	--------------------	-----------	-------------------



- Bước 1: Từ giao diện chính của RethinkDB ấn vào mục Tables
- Bước 2: Chọn database có bảng cần xóa, click vào bảng cần xóa trong database đó
- Bước 3: Click vào nút Operation, chọn Delete table
- Bước 4: Click vào Delete để xóa bảng

2.5.5. Thêm dữ liệu vào bảng

CÚ PHÁP: `r.db('<TÊN DATABASE>').table('<TÊN BẢNG>').insert(<document>)`

SQL	ReQL
INSERT INTO CHINHANH VALUES ('CN01', 'Jiangnam khu A');	<code>r.db('JIANGNAM_BAKERY').table('CHINHANH').insert({MACN: "CN01", TENCN: "Jiangnam khu A"});</code>

The screenshot shows the Data Explorer interface. At the top, there's a header with 'Data Explorer' and buttons for 'History', settings, and a refresh icon. Below the header is a code editor with a single line of SQL: `1 r.db('JIANGNAM_BAKERY').table('CHINHANH').insert({MACN: "CN01", TENCN: "Jiangnam khu A"});`. To the right of the code editor are 'Clear' and 'Run' buttons. Below the code editor, it says '1 row returned in 6ms.' 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 dữ liệu cùng lúc:

The screenshot shows the Data Explorer interface. At the top, there's a header with 'Data Explorer' and buttons for 'History', settings, and a refresh icon. Below the header is a code editor with a bulk insert query:

```
1 r.db('JIANGNAM_BAKERY').table('CHINHANH').insert([
2   {MACN: "CN01", TENCN: "Jiangnam khu A"},
3   {MACN: "CN02", TENCN: "Jiangnam khu B"},
4   {MACN: "CN03", TENCN: "Jiangnam Thu Duc"}
5 ]);
```

 To the right of the code editor are 'Clear' and 'Run' buttons. Below the code editor, it says '1 row returned in 9ms.' 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": 3 ,  "replaced": 0 ,  "skipped": 0 ,  "unchanged": 0 }
```

Đối số truyền vào là một mảng chứa 03 đối tượng, mỗi đối tượng tương ứng một bản ghi mới. Như vậy sau khi thực hiện câu lệnh trên, có 3 bản ghi mới được thêm vào bảng CHINHANH.

2.5.6. Truy vấn dữ liệu

a) Thao tác SELECT

SQL	ReQL
SELECT * FROM NHANVIEN	r.table("NHANVIEN")
SELECT MASO_NV, TEN_NV FROM NHANVIEN	r.table("NHANVIEN") .pluck("MASO_NV", "TEN_NV")
SELECT * FROM NHANVIEN WHERE TEN_NV = "NGUYEN THY ANH TAI"	r.table("NHANVIEN").filter({ TEN_NV: "NGUYEN THY ANH TAI" })
SELECT * FROM NHANVIEN WHERE TEN_NV = "NGUYEN ANH PHI" AND LUONG = 5000000	r.table("NHANVIEN").filter({ TENNV: "NGUYEN ANH PHI", LUONG: 5000000 })
SELECT MASO_NV FROM NHANVIEN WHERE TEN_NV = "ANH TAI" ORDER BY TENNV DESC	r.table("NHANVIEN").filter({ TEN_NV: "ANH TAI" }).orderBy(r.desc("TEN_NV")).pluck("MASO_NV")
SELECT COUNT(*) FROM NHANVIEN	r.table("NHANVIEN").count()
SELECT COUNT(MANV) FROM NHANVIEN WHERE LUONG > 1000000	r.table("NHANVIEN").filter(r.row.hasFields("MASO_NV") .and(r.row("LUONG").gt(1000000))).count()

SELECT AVG("LUONG") FROM NHANVIEN	r.table("NHANVIEN").avg("LUONG")
SELECT MAX("LUONG") FROM NHANVIEN	r.table("NHANVIEN")("LUONG").max())
SELECT DISTINCT(TEN_NV) FROM NHANVIEN	r.table("NHANVIEN").pluck("TEN_NV")) .distinct()
SELECT * FROM NHANVIEN WHERE LUONG BETWEEN 10000000 AND 15000000;	r.table("NHANVIEN").filter(r.row("LUONG").ge(10000000) .and(r.row("LUONG").le(15000000)))
SELECT * FROM NHANVIEN ORDER BY LUONG DESC LIMIT 5;	r.table("NHANVIEN") .orderBy(r.desc("LUONG")) .limit(5)

Một số minh họa:

- Chọn tất cả nhân viên trong bảng NHANVIEN, tương đương `SELECT * FROM NHANVIEN`:

The screenshot shows the Data Explorer interface. The query editor contains the following SQL query:

```
1 r.db('Jiangnam1').table('NHANVIEN')
```

Buttons for 'Clear' and 'Run' are visible. Below the query, it states '4 rows returned. Displaying rows 0-3'. The 'Table view' tab is selected, displaying the following table:

	GIOITINH	LUONG	MASO_CN	MASO_NV	TEN_NV
0	Nam	3000000	CN1	NV03	Dang Nhat Nam
1	Nu	6000000	CN1	NV04	Bui My Linh
2	Nam	4050000	CN1	NV02	Tran Anh Khoi
3	Nu	5000000	CN1	NV01	Le Thi Hanh

- Chọn mức lương cao nhất của nhân viên, tương đương `SELECT MAX("LUONG") FROM NHANVIEN`

The screenshot shows the Data Explorer interface. The query editor contains the following SQL query:

```
1 r.db('Jiangnam1').table('NHANVIEN')("LUONG").max()
```

Buttons for 'Clear' and 'Run' are visible. Below the query, it states '1 row returned in 3ms.'. The 'Table view' tab is selected, displaying the following table:

	value
0	6000000

- Sử dụng hàm `pluck()` để chọn một số thuộc tính nhất định.
VD: Chọn nhân viên (`MASO_NV`, `TEN_NV`) từ bảng `NHANVIEN`, tương đương `SELECT MANV, TENNV FROM NHANVIEN`

Data Explorer History Settings Fullscreen

```
1 r.db('Jiangnam1').table('NHANVIEN').pluck('MASO_NV', 'TEN_NV')
2
```

Clear Run

4 rows returned. Displaying rows 0-3

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	MASO_NV	TEN_NV
0	NV03	Dang Nhat Nam
1	NV04	Bui My Linh
2	NV02	Tran Anh Khoi
3	NV01	Le Thi Hanh

- Ngoài ra, còn có thể lấy dữ liệu bằng cách sử dụng hàm `.get()` với tham số truyền vào là giá trị khóa chính.

Data Explorer History Settings Fullscreen

```
1 r.db('Jiangnam1').table('CHINHANH').get('CN1')
```

Clear Run

1 row returned in 1ms.

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	DIACHI	MASO_CN	TEN_CN
0	KP6, Di An, Binh Duong	CN1	Jiangnam bakery Lang dai hoc

- Sử dụng hàm `filter()` để thể hiện mệnh đề WHERE, có thể kết hợp thêm hàm `and()`, `eq()`, `gt()`, `ge()`,...

VD: Chọn nhân viên có giới tính là nam và có mức lương nhiều hơn 4 triệu, tương đương `SELECT * FROM NHANVIEN WHERE LUONG>4000000 AND GIOITINH= 'Nam';`

Data Explorer History Settings Fullscreen

```
1 r.db('Jiangnam1').table('NHANVIEN').filter(
2   r.row('LUONG').gt(4000000)
3   .and(r.row('GIOITINH').eq('Nam'))
4 )
5
```

Clear Run

1 row returned.

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	GIOITINH	LUONG	MASO_CN	MASO_NV	TEN_NV
0	Nam	4050000	CN1	NV02	Tran Anh Khoi

- Sử dụng hàm `innerJoin()`, `eqJoin()` và `zip()` để kết bảng:

VD: Chọn nhân viên và chi nhánh làm việc của nhân viên đó, tương đương

`SELECT * FROM NHANVIEN NV JOIN CHINHANH CN ON NV.MASO_CN = CN.MASO_CN;`

Data Explorer History Settings Help

```

1 r.db('Jiangnam1').table('NHANVIEN').innerJoin(
2   r.db('Jiangnam1').table('CHINHANH'),
3   function (nhanvien, chinhanh) {
4     return nhanvien('MASO_CN').eq(chinhanh('MASO_CN'));
5   }).zip()

```

Clear Run

4 rows returned. Displaying rows 0-3

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	DIACHI	GIOITINH	LUONG	MASO_CN	MASO_NV	TEN_CN	TEN_NV
0	KP6, Di An, Binh Duong	Nam	3000000	CN1	NV03	Jiangnam bakery Lang dai hoc	Dang Nhat Nam
1	KP6, Di An, Binh Duong	Nu	6000000	CN1	NV04	Jiangnam bakery Lang dai hoc	Bui My Linh
2	KP6, Di An, Binh Duong	Nam	4050000	CN1	NV02	Jiangnam bakery Lang dai hoc	Tran Anh Khoi
3	KP6, Di An, Binh Duong	Nu	5000000	CN1	NV01	Jiangnam bakery Lang dai hoc	Le Thi Hanh

Nếu như bảng được kết (right table) được đánh chỉ mục là khóa chính (primary key) hay có chỉ mục thứ cấp (secondary index) thì có thể sử dụng hàm `eqJoin()` để tăng hiệu suất.

Data Explorer History Settings Help

```

1 r.db('Jiangnam1').table('NHANVIEN').eqJoin(
2   "MASO_CN",
3   r.db('Jiangnam1').table('CHINHANH'),
4   {index: "MASO_CN"}
5 ).zip()

```

Clear Run

4 rows returned. Displaying rows 0-3

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	DIACHI	GIOITINH	LUONG	MASO_CN	MASO_NV	TEN_CN	TEN_NV
0	KP6, Di An, Binh Duong	Nam	3000000	CN1	NV03	Jiangnam bakery Lang dai hoc	Dang Nhat Nam
1	KP6, Di An, Binh Duong	Nu	6000000	CN1	NV04	Jiangnam bakery Lang dai hoc	Bui My Linh
2	KP6, Di An, Binh Duong	Nam	4050000	CN1	NV02	Jiangnam bakery Lang dai hoc	Tran Anh Khoi
3	KP6, Di An, Binh Duong	Nu	5000000	CN1	NV01	Jiangnam bakery Lang dai hoc	Le Thi Hanh

- Sử dụng hàm group(), ungroup() cho việc gom nhóm và các hàm count(), sum(),... cho việc tính toán.

VD: Cho biết các LOẠI sản phẩm và số lượng sản phẩm thuộc các loại đó, tương đương SELECT LOAI, COUNT(MASO_SP) FROM SANPHAM GROUP BY LOAI, COUNT(MASO_SP);

Data Explorer

```

1 r.db('Jiangnam1').table('SANPHAM')
2 .group('LOAI')
3 .count()

```

3 rows returned in 1ms.

Tree view Table Viewer View Table view Raw view Query profile

	group	reduction
0	Do an	5
1	Thuc uong	2
2	Trang mieng	3

b) Thao tác UPDATE

- Sử dụng hàm update() cho việc thay đổi giá trị dữ liệu, có thể dùng thêm hàm add() để cộng thêm, sub() để trừ, mul() để nhân,... các giá trị dữ liệu.

VD: Cập nhật tăng lương thêm 20% cho nhân viên có mã số nhân viên là NV03, tương đương UPDATE NHANVIEN SET LUONG=LUONG*1.2 WHERE MASO_NV='NV03'

Data Explorer

```

1 r.db('Jiangnam1').table('NHANVIEN').get('NV03').update(
2 {LUONG: r.row('LUONG').mul(1.2)}
3 )

```

1 row returned in 7ms.

Tree view Table Viewer View Table view Raw view Query profile

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

c) Thao tác DELETE

- Sử dụng hàm delete() để xóa dữ liệu của bảng. Có thể xóa tất cả dữ liệu của bảng hoặc xóa có điều kiện.

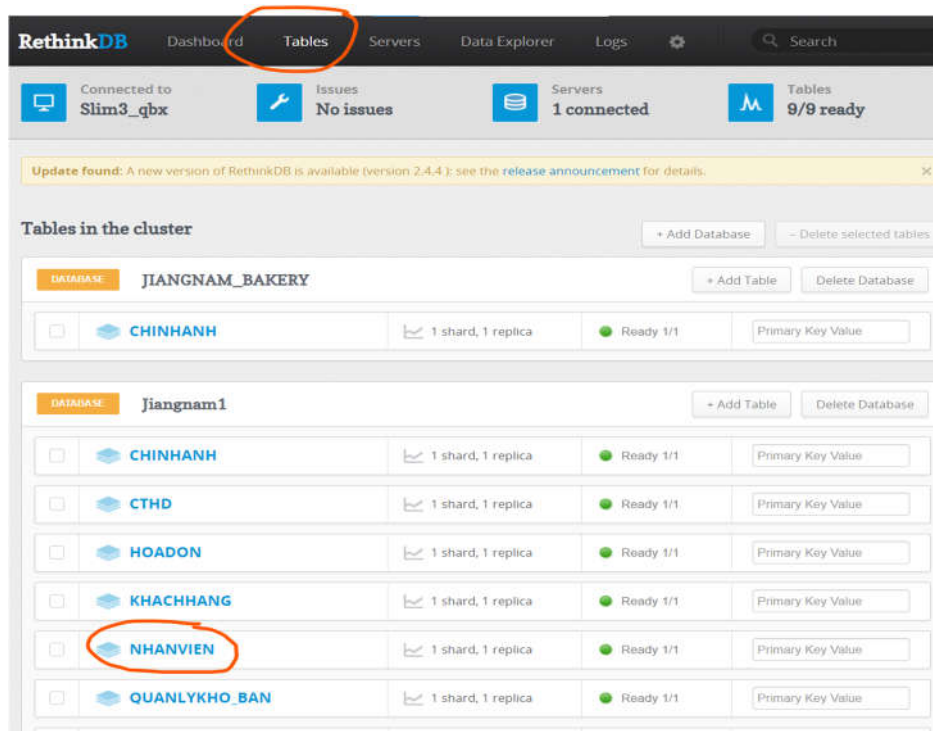
VD: Xóa các sản phẩm có loại là 'Trang mieng', tương đương DELETE FROM SANPHAM WHERE LOAI = 'Trang mieng'



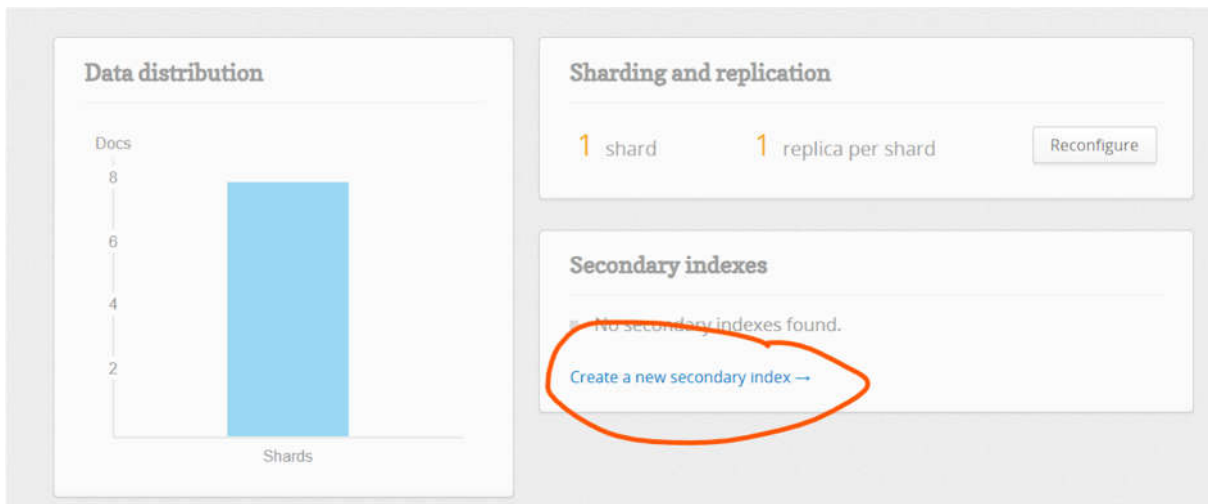
d) Chỉ mục thứ cấp (secondary indexes)

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 các loại chỉ mục thứ cấp (secondary indexes) như: chỉ mục đơn (simple indexes) dựa trên giá trị của một trường dữ liệu, chỉ mục phức hợp (compound indexes) dựa trên nhiều trường dữ liệu, đa chỉ mục (multi indexes) dựa trên một mảng các giá trị và chỉ mục dựa trên biểu thức phân cấp (arbitrary expressions). Dưới đây là cách tạo chỉ mục đơn (simple indexes) cho bảng dữ liệu.

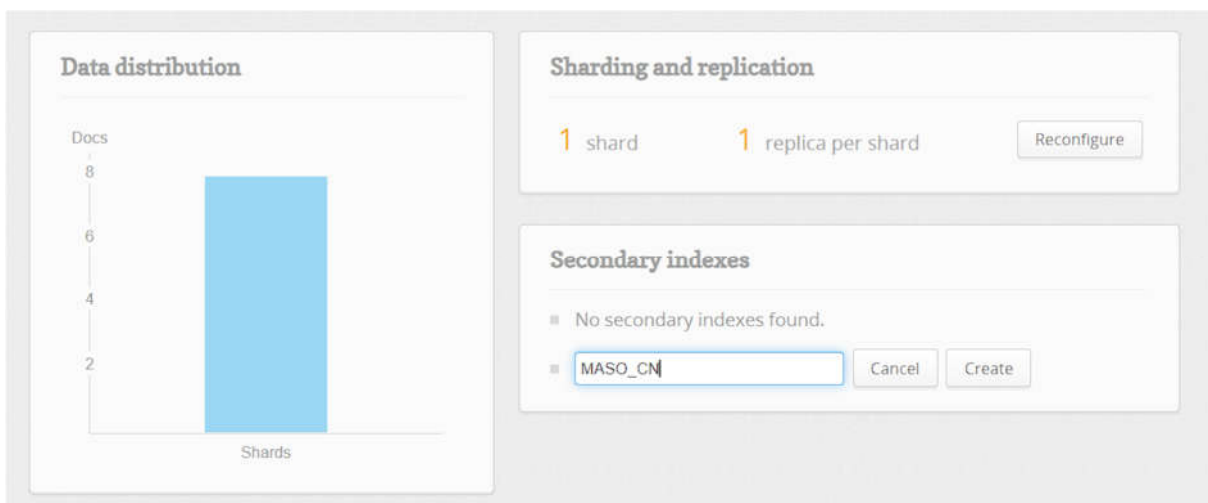
- Bước 1: Tại màn hình RethinkDB Administration Console, truy cập thẻ Tables, chọn bảng cần tạo chỉ mục thứ cấp. (VD ở đây là bảng NHANVIEN)



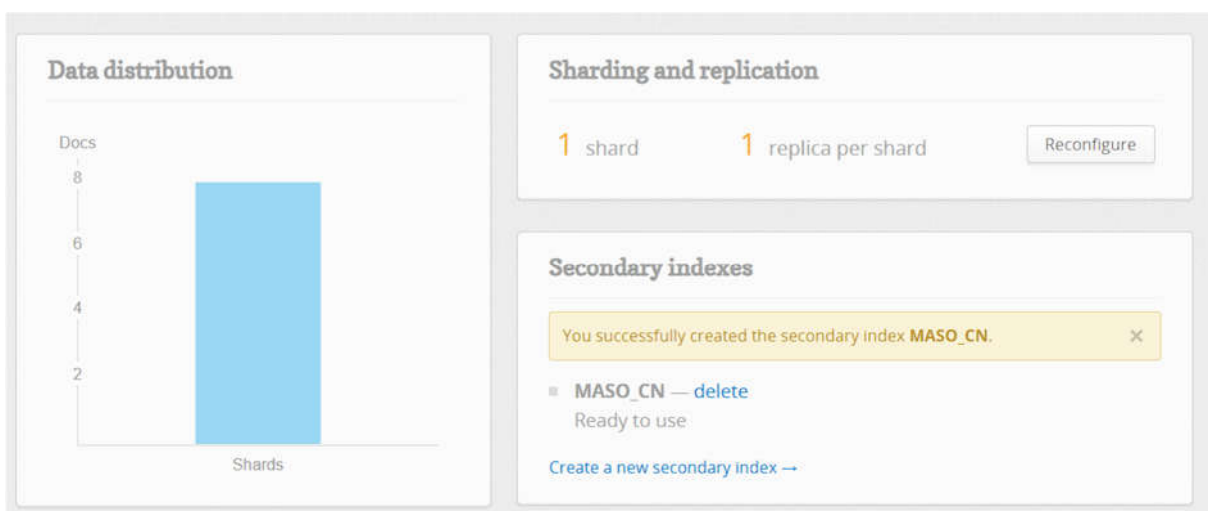
- Bước 2: Tại phần Secondary indexes, nhấn chọn “Create a new secondary index”



- Bước 3: Nhập tên chỉ mục thứ cấp và nhấn Create

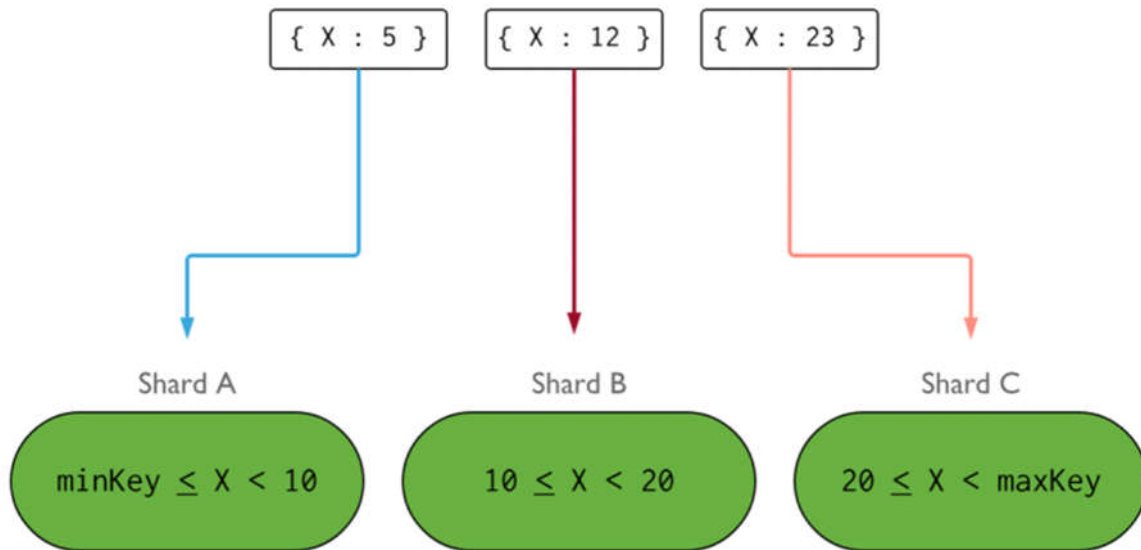


- Bước 4: Chỉ mục thứ cấp đã sẵn sàng sử dụng, nếu muốn ngừng sử dụng thì chọn delete và xác nhận xóa.



2.6 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.

Servers used by this table

Shard 1 ~10 documents

DESKTOP_TO11UMB_J20 Primary replica ready

Table Viewer

Lookup: key

id	MASO_CN	MASO_SP	NGAYNHAP	SOLUONGNHAP
7aba6811-c54...	CN1	SP09	01/12/2023	57
a3137489-d31...	CN1	SP07	01/12/2023	10
b9844263-0d3...	CN1	SP05	01/12/2023	114
d0a37292-a4a...	CN1	SP01	01/12/2023	88
d902ad96-e09...	CN1	SP02	01/12/2023	100
de598cd0-029...	CN1	SP03	01/12/2023	123
e9b18090-ef6...	CN1	SP04	01/12/2023	198

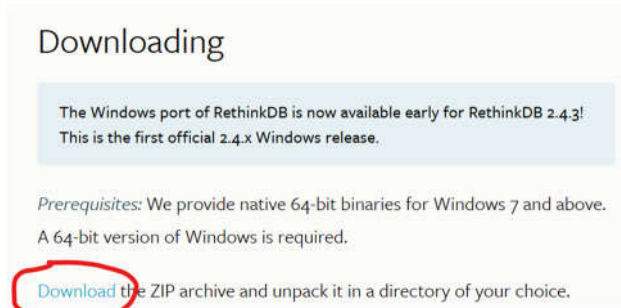
End of results

Ví dụ: bảng QUANLYKHO_NHAP và dữ liệu được tạo và nhập trên máy 1, nhưng RethinkDB tự lưu dữ liệu của bảng này trên máy 2.

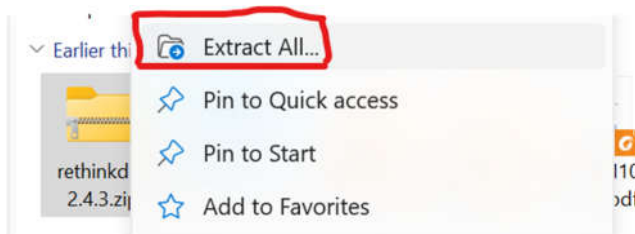
CHƯƠNG 2: CÀI ĐẶT VÀ KẾT NỐI NHIỀU MÁY

1. Cài đặt RethinkDB trên Windows

- Truy cập vào đường link: [Install RethinkDB on Windows - RethinkDB](#)
- Ở phần Downloading click vào Download



- Sau khi tải xong, giải nén file .zip vừa tải xuống



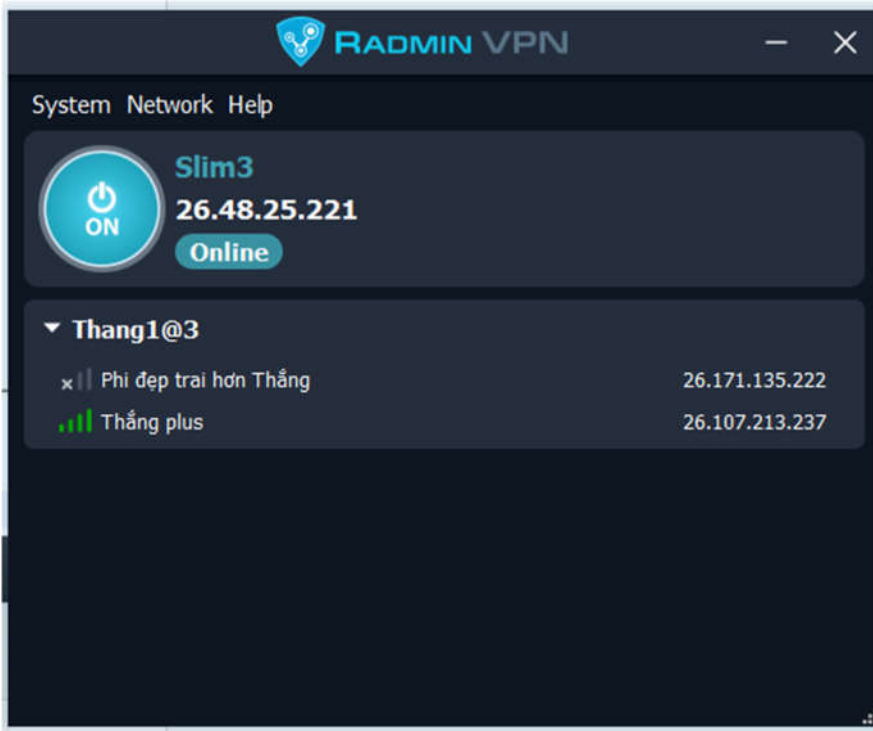
- Để khởi động RethinkDB ta dùng cmd hoặc terminal trở đến thư mục vừa giải nén và chạy lệnh:

Rethinkdb.exe

```
C:\Users\THANG\Downloads\rethinkdb-2.4.3\rethinkdb-2.4.3>rethinkdb.exe
Running rethinkdb 2.4.2-11-g6613dc (Windows) (MSC 191627048)...
Running on 10.0.22621 (Windows 10, Server 2016)
Loading data from directory C:\Users\THANG\Downloads\rethinkdb-2.4.3\rethinkdb-2.4.3\rethinkdb_data
warn: Cache size does not leave much memory for server and query overhead (available memory: 875 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 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
option.
Server ready, "DESKTOP_T011UMB_omn" 7ac0d467-1677-43c6-a29a-c8216500d5bb
```

- Ở đây chúng ta có 3 cổng intracluster, client driver, và administrative HTTP và các địa chỉ của chúng.
 - + Intracluster: Là cổng để các server kết nối với nhau
 - + Client driver: Là cổng để phần mềm chương trình kết nối với server
 - + Administrative HTTP: Là cổng truy cập đến giao diện web của RethinkDB
- Để truy cập đến giao diện web của RethinkDB, chúng ta mở trình duyệt web và nhập vào địa chỉ: *127.0.0.1:8080*

2. Kết nối các máy 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 Radmin VPN (đã được học trong chương trình)
- Bước 1: Đảm bảo các máy đã có chung network Radmin VPN, đảm bảo terminal đã cd sang thư mục chứa rethinkdb.exe
- Bước 2: Máy chủ chạy lệnh `rethinkdb --bind all` để chạy server máy chủ của mình trên mạng LAN ảo.
- Bước 3: Các máy trạm chạy lệnh `rethinkdb --directory rethinkdb_data2 --join <Địa chỉ IP máy chủ>`
- Bước 4: Vào tab server của RethinkDB để xem các server kết nối với nhau.

CHƯƠNG 3: TRUY VẤN TRÊN MÔI TRƯỜNG PHÂN TÁN

1. Nhập liệu trên từng máy

1.1 Nhập liệu trên máy 1

- Tạo database Jiangnam1

The screenshot shows the Data Explorer interface. At the top, there's a toolbar with 'History', 'Settings', and 'Full Screen' icons. Below it, a code editor contains the SQL command: `1 r.dbCreate('Jiangnam1');`. To the right of the code editor are 'Clear' and 'Run' buttons. Below the code editor, it says '1 row returned in <1ms.' 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 response:

```
{  "config_changes": [    {      "new_val": {        "id": "15369880-a000-4593-9047-bcd44a825291",        "name": "Jiangnam1"      },      "old_val": null    }  ],  "dbs_created": 1}
```

- Tạo các bảng dữ liệu

The screenshot shows the Data Explorer interface. At the top, there's a toolbar with 'History', 'Settings', and 'Full Screen' icons. Below it, a code editor contains the following SQL commands:

```
1 r.db('Jiangnam1').tableCreate('CHINHANH', {primaryKey: "MASO_CN"});
2 r.db('Jiangnam1').tableCreate('NHANVIEN', {primaryKey: "MASO_NV"});
3 r.db('Jiangnam1').tableCreate('SANPHAM', {primaryKey: "MASO_SP"});
4 r.db('Jiangnam1').tableCreate('KHACHHANG', {primaryKey: "MASO_KH"});
5 r.db('Jiangnam1').tableCreate('QUANLYKHO_NHAP', {primaryKey: "MASO_CN"});
6 r.db('Jiangnam1').tableCreate('QUANLYKHO_BAN', {primaryKey: "MASO_CN"});
7 r.db('Jiangnam1').tableCreate('HOADON', {primaryKey: "MASO_HD"});
8 r.db('Jiangnam1').tableCreate('CTHD', {primaryKey: "MASO_HD"});
9
10
```

 To the right of the code editor are 'Clear' and 'Run' buttons. Below the code editor, it says '1 row returned in 549ms.' and there are tabs for 'Tree view', 'Table Viewer View', 'Table view', 'Raw view', and 'Query profile'.

- Nhập dữ liệu bảng CHINHANH

Data Explorer

```

1  r.db('Jiangnam1').table('CHINHANH').insert(
2  [
3  {
4      MASO_CN: 'CN1',
5      TEN_CN: 'Jiangnam bakery Lang dai hoc',
6      DIACHI: 'KP6, Di An, Binh Duong'
7  },
8  {
9      MASO_CN: 'CN2',
10     TEN_CN: 'Jiangnam bakery An Phu',
11     DIACHI: '191 Le Thanh Tong, Quan 2, TP HCM'
12  }
13  ]
14  );
15

```

- Nhập dữ liệu bảng NHANVIEN

```

17  r.db('Jiangnam1').table('NHANVIEN').insert(
18  [
19  {
20     MASO_NV: 'NV01',
21     TEN_NV: 'Le Thi Hanh',
22     GIOITINH: 'Nu',
23     LUONG: 5000000,
24     MASO_CN: 'CN1'
25  },
26  {
27     MASO_NV: 'NV02',
28     TEN_NV: 'Tran Anh Khoi',
29     GIOITINH: 'Nam',
30     LUONG: 4050000,
31     MASO_CN: 'CN1'
32  },
33  {
34     MASO_NV: 'NV03',
35     TEN_NV: 'Dang Nhat Nam',
36     GIOITINH: 'Nam',
37     LUONG: 3000000,
38     MASO_CN: 'CN1'
39  },
40  {
41     MASO_NV: 'NV04',
42     TEN_NV: 'Bui My Linh',
43     GIOITINH: 'Nu',
44     LUONG: 6000000,
45     MASO_CN: 'CN1'
46  }
47  ]
48  );

```

- Nhập dữ liệu bảng KHACHHANG

```
50 r.db('Jiangnam1').table('KHACHHANG').insert(  
51 [{  
52   MASO_KH: 'KH01',  
53   TEN_KH: 'Nguyen Anh Tai',  
54   DIACHI: 'KTX Khu A, KP6, Linh Trung, Thu Duc, TP HCM',  
55   SDT: '098675849',  
56   NGAYSINH: '15/06/2002',  
57   NGAY_DK: '25/01/2020',  
58   LOAI: 'DONG'  
59 }],  
60 {  
61   MASO_KH: 'KH02',  
62   TEN_KH: 'Chu Huyen Trang',  
63   DIACHI: '245 Dang Van Bi, Thu Duc, TP HCM',  
64   SDT: '0984374635',  
65   NGAYSINH: '19/01/1993',  
66   NGAY_DK: '21/04/2020',  
67   LOAI: 'DONG'  
68 },  
69 {  
70   MASO_KH: 'KH03',  
71   TEN_KH: 'Nguyen Phuc Hao',  
72   DIACHI: '199 An Phu Thinh, Quan 2, TP HCM',  
73   SDT: '0733456778',  
74   NGAYSINH: '11/05/2000',  
75   NGAY_DK: '18/10/2020',  
76   LOAI: 'DONG'  
77 },  
78 {  
79   MASO_KH: 'KH04',  
80   TEN_KH: 'Nguyen Xuan Thi',  
81   DIACHI: '18 Hong Linh, An Nhon, Binh Dinh',  
82   SDT: '0987654321',  
83   NGAYSINH: '15/08/1991',  
84   NGAY_DK: '12/08/2022',  
85   LOAI: 'DONG'  
86 },  
87 {  
88   MASO_KH: 'KH05',  
89   TEN_KH: 'Ngo Chan Huy',  
90   DIACHI: '171 Huynh Thuc Khang, Q4, TP HCM',  
91   SDT: '0879654378',  
92   NGAYSINH: '25/06/1999',  
93   NGAY_DK: '25/08/2023',  
94   LOAI: 'DONG'  
95 }}, {
```

- Nhập dữ liệu bảng SANPHAM

```
161 r.db('Jiangnam1').table('SANPHAM').insert(  
162 [  
163   {  
164     "MASO_SP": "SP01",  
165     "TEN_SP": "Oreo",  
166     "QUOCCGIA": "Phap",  
167     "GIAGOC": 25000,  
168     "GIABAN": 35000,  
169     "LOAI": "Do an",  
170     "VAT": 10  
171   },  
172   {  
173     "MASO_SP": "SP02",  
174     "TEN_SP": "Kitkat",  
175     "QUOCCGIA": "Nhat Ban",  
176     "GIAGOC": 10000,  
177     "GIABAN": 45000,  
178     "LOAI": "Do an",  
179     "VAT": 10  
180   },  
181   {  
182     "MASO_SP": "SP03",  
183     "TEN_SP": "Tiramisu",  
184     "QUOCCGIA": "Anh",  
185     "GIAGOC": 15000,  
186     "GIABAN": 30000,  
187     "LOAI": "Trang mieng",  
188     "VAT": 10  
189   },  
190   {  
191     "MASO_SP": "SP04",  
192     "TEN_SP": "Lamington",  
193     "QUOCCGIA": "Viet Nam",  
194     "GIAGOC": 5000,  
195     "GIABAN": 10000,  
196     "LOAI": "Trang mieng",  
197     "VAT": 10  
198   },  
199   {  
200     "MASO_SP": "SP05",  
201     "TEN_SP": "Tobboki",  
202     "QUOCCGIA": "Han Quoc",  
203     "GIAGOC": 15000,  
204     "GIABAN": 20000,  
205     "LOAI": "Do an",  
206     "VAT": 10  
207   },  
208   {
```


- Nhập dữ liệu bảng QUANLYKHO_NHAP

```
257 r.db('Jiangnam1').table('QUANLYKHO_NHAP').insert(  
258 [  
259   {  
260     "MASO_CN": "CN1",  
261     "MASO_SP": "SP01",  
262     "NGAYNHAP": "01/12/2023",  
263     "SOLUONGNHAP": 88  
264   },  
265   {  
266     "MASO_CN": "CN1",  
267     "MASO_SP": "SP02",  
268     "NGAYNHAP": "01/12/2023",  
269     "SOLUONGNHAP": 100  
270   },  
271   {  
272     "MASO_CN": "CN1",  
273     "MASO_SP": "SP03",  
274     "NGAYNHAP": "01/12/2023",  
275     "SOLUONGNHAP": 123  
276   },  
277   {  
278     "MASO_CN": "CN1",  
279     "MASO_SP": "SP04",  
280     "NGAYNHAP": "01/12/2023",  
281     "SOLUONGNHAP": 198  
282   },  
283   {  
284     "MASO_CN": "CN1",  
285     "MASO_SP": "SP05",  
286     "NGAYNHAP": "01/12/2023",  
287     "SOLUONGNHAP": 114  
288   },  
289   {  
290     "MASO_CN": "CN1",  
291     "MASO_SP": "SP07",  
292     "NGAYNHAP": "01/12/2023",  
293     "SOLUONGNHAP": 10  
294   },  
295   {  
296     "MASO_CN": "CN1",  
297     "MASO_SP": "SP09",  
298     "NGAYNHAP": "01/12/2023",  
299     "SOLUONGNHAP": 57  
300   }  
301 ]
```

- Nhập dữ liệu bảng QUANLYKHO_BAN

```
304 r.db('Jiangnam1').table('QUANLYKHO_BAN').insert(  
305 [  
306   {  
307     "MASO_CN": "CN1",  
308     "MASO_SP": "SP01",  
309     "TRANGTHAI": "Dang kinh doanh",  
310     "SOLUONGTON": 88  
311   },  
312   {  
313     "MASO_CN": "CN1",  
314     "MASO_SP": "SP02",  
315     "TRANGTHAI": "Ngung kinh doanh",  
316     "SOLUONGTON": 0  
317   },  
318   {  
319     "MASO_CN": "CN1",  
320     "MASO_SP": "SP03",  
321     "TRANGTHAI": "Dang nhap hang",  
322     "SOLUONGTON": 123  
323   },  
324   {  
325     "MASO_CN": "CN1",  
326     "MASO_SP": "SP04",  
327     "TRANGTHAI": "Dang kinh doanh",  
328     "SOLUONGTON": 198  
329   },  
330   {  
331     "MASO_CN": "CN1",  
332     "MASO_SP": "SP05",  
333     "TRANGTHAI": "Ngung kinh doanh",  
334     "SOLUONGTON": 25  
335   },  
336   {  
337     "MASO_CN": "CN1",  
338     "MASO_SP": "SP06",  
339     "TRANGTHAI": "Dang nhap hang",  
340     "SOLUONGTON": 114  
341   },  
342   {  
343     "MASO_CN": "CN1",  
344     "MASO_SP": "SP07",  
345     "TRANGTHAI": "Dang kinh doanh",  
346     "SOLUONGTON": 8  
347   },  
348   {  
349     "MASO_CN": "CN1",  
350     "MASO_SP": "SP08",  
351     "TRANGTHAI": "Ngung kinh doanh",  
352     "SOLUONGTON": 0  
353   },  
354 ])
```

- Nhập dữ liệu bảng HOADON

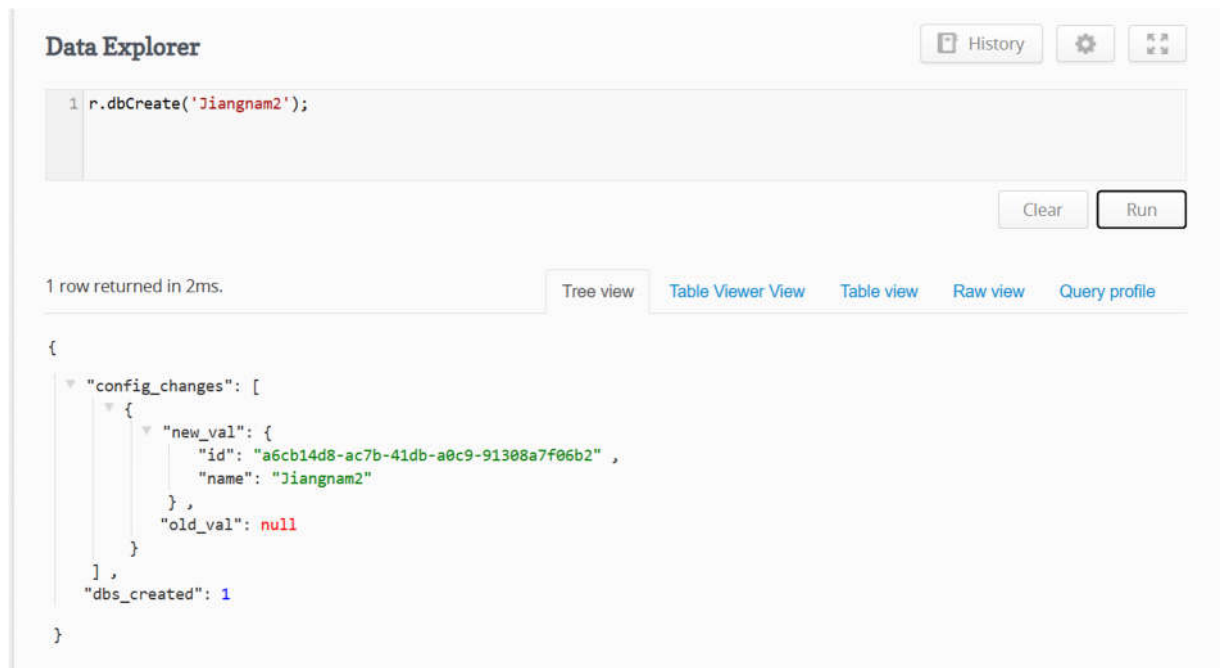
```
369 r.db('Jiangnam1').table('HOADON').insert(  
370 [  
371   {  
372     "MASO_HD": "HD01",  
373     "MASO_NV": "NV01",  
374     "MASO_KH": "KH01",  
375     "MASO_CN": "CN1",  
376     "NGAYXUAT": "02/11/2025",  
377     "TONGTIEN": 0  
378   },  
379   {  
380     "MASO_HD": "HD02",  
381     "MASO_NV": "NV02",  
382     "MASO_KH": "KH01",  
383     "MASO_CN": "CN1",  
384     "NGAYXUAT": "02/01/2023",  
385     "TONGTIEN": 0  
386   },  
387   {  
388     "MASO_HD": "HD03",  
389     "MASO_NV": "NV02",  
390     "MASO_KH": "KH02",  
391     "MASO_CN": "CN1",  
392     "NGAYXUAT": "22/01/2024",  
393     "TONGTIEN": 0  
394   },  
395   {  
396     "MASO_HD": "HD04",  
397     "MASO_NV": "NV01",  
398     "MASO_KH": "KH02",  
399     "MASO_CN": "CN1",  
400     "NGAYXUAT": "02/11/2024",  
401     "TONGTIEN": 0  
402   },  
403   {  
404     "MASO_HD": "HD05",  
405     "MASO_NV": "NV01",  
406     "MASO_KH": "KH03",  
407     "MASO_CN": "CN1",  
408     "NGAYXUAT": "02/02/2024",  
409     "TONGTIEN": 0  
410   }  
411 ])
```

- Nhập dữ liệu bảng CTHD

```
414 r.db('Jiangnam1').table('CTHD').insert(  
415  
416 [  
417   {  
418     "MASO_HD": "HD01",  
419     "MASO_SP": "SP01",  
420     "SOLUONG": 3  
421   },  
422   {  
423     "MASO_HD": "HD01",  
424     "MASO_SP": "SP02",  
425     "SOLUONG": 2  
426   },  
427   {  
428     "MASO_HD": "HD01",  
429     "MASO_SP": "SP06",  
430     "SOLUONG": 2  
431   },  
432   {  
433     "MASO_HD": "HD02",  
434     "MASO_SP": "SP01",  
435     "SOLUONG": 3  
436   },  
437   {  
438     "MASO_HD": "HD02",  
439     "MASO_SP": "SP02",  
440     "SOLUONG": 2  
441   },  
442   {  
443     "MASO_HD": "HD03",  
444     "MASO_SP": "SP01",  
445     "SOLUONG": 3  
446   },  
447   {  
448     "MASO_HD": "HD03",  
449     "MASO_SP": "SP02",  
450     "SOLUONG": 2  
451   },  
452   {  
453     "MASO_HD": "HD03",  
454     "MASO_SP": "SP06",  
455     "SOLUONG": 2  
456   },  
457   {  
458     "MASO_HD": "HD04",  
459     "MASO_SP": "SP04",  
460     "SOLUONG": 3  
461   },  
462 ]
```

1.2 Nhập liệu trên máy 2

- Tạo database Jiangnam2



- Tạo các bảng dữ liệu



- Nhập dữ liệu cho bảng CHINHANH

Data Explorer History Settings Fullscreen

```

1 r.db('Jiangnam1').table('CHINHANH').insert(
2
3 [
4
5 {
6
7     MASO_CN: 'CN1',
8
9     TEN_CN: 'Jiangnam bakery Lang dai hoc',
10
11     DIACHI: 'KP6, Di An, Binh Duong'
12 },
13 ],
14
15 {
16
17     MASO_CN: 'CN2',
18
19     TEN_CN: 'Jiangnam bakery An Phu',
20
21     DIACHI: '191 Le Thanh Tong, Quan 2, TP HCM'
22 }
23 ]
24 ]
25 ]
26 ]
27 );

```

Clear Run

- Nhập dữ liệu bảng NHANVIEN

```

1 r.db('Jiangnam2').table('NHANVIEN').insert(
2
3 [
4
5 {
6
7     "MASO_NV": "NV05",
8
9     "TEN_NV": "Nguyen Minh Thu",
10
11     "GIOITINH": "Nu",
12
13     "LUONG": 7500000,
14
15     "MASO_CN": "CN2"
16 },
17 ],
18
19 {
20
21     "MASO_NV": "NV06",
22
23     "TEN_NV": "Phan Bich Ngoc",
24
25     "GIOITINH": "Nu",
26
27     "LUONG": 4500000,
28
29     "MASO_CN": "CN2"
30 },
31 ],
32
33 {
34
35     "MASO_NV": "NV07",
36
37     "TEN_NV": "Thai Minh Tri",
38
39     "GIOITINH": "Nam",
40
41     "LUONG": 5500000,
42
43     "MASO_CN": "CN2"
44 },
45 ],
46
47 {
48
49     "MASO_NV": "NV08",
50
51     "TEN_NV": "Dao Cong Tuan",
52
53     "GIOITINH": "Nam",
54
55     "LUONG": 6000000,
56
57     "MASO_CN": "CN2"
58 }
59 ]
60 ]
61 ]
62 ]
63 );

```

- Nhập dữ liệu bảng KHACHHANG

Data Explorer

```

1 r.db('Jiangnam1').table('KHACHHANG').insert(
2
3 [{
4
5   MASO_KH: 'KH01',
6
7   TEN_KH: 'Nguyen Anh Tai',
8
9   DIACHI: 'KTX Khu A, KP6, Linh Trung, Thu Duc, TP HCM',
10
11   SDT: '098675849',
12
13   NGAYSINH: '15/06/2002',
14
15   NGAY_DK: '25/01/2020',
16
17   LOAI: 'DONG'
18
19 },
20
21 {
22
23   MASO_KH: 'KH02',
24
25   TEN_KH: 'Chu Huyen Trang',
26
27   DIACHI: '245 Dang Van Bi, Thu Duc, TP HCM',
28
29   SDT: '0984374635',
30
31   NGAYSINH: '19/01/1993',
32
33   NGAY_DK: '21/04/2020',
34
35   LOAI: 'DONG'
36
37 },
38
39 {
40
41   MASO_KH: 'KH03',
42
43   TEN_KH: 'Nguyen Phuc Hao',
44
45   DIACHI: '199 An Phu Thinh, Quan 2, TP HCM',
46
47   SDT: '0733456778',
48
49   NGAYSINH: '11/05/2000',
50
51   NGAY_DK: '18/10/2020',
52
53   LOAI: 'DONG'
54
55 },
56
57 {
58
59   MASO_KH: 'KH04',
60
61   TEN_KH: 'Nguyen Xuan Thi',
62
63   DIACHI: '18 Hong Linh, An Nhon, Binh Dinh',
64
65   SDT: '0987654321',
66

```

- Nhập dữ liệu bảng SANPHAM

Data Explorer

```
1 r.db('Jiangnam1').table('SANPHAM').insert(  
2  
3 [  
4  
5   {  
6  
7     "MASO_SP": "SP01",  
8  
9     "TEN_SP": "Oreo",  
10  
11     "QUOCCGIA": "Phap",  
12  
13     "GIAGOC": 25000,  
14  
15     "GIABAN": 35000,  
16  
17     "LOAI": "Do an",  
18  
19     "VAT": 10  
20  
21   },  
22  
23   {  
24  
25     "MASO_SP": "SP02",  
26  
27     "TEN_SP": "Kitkat",  
28  
29     "QUOCCGIA": "Nhat Ban",  
30  
31     "GIAGOC": 10000,  
32  
33     "GIABAN": 45000,  
34  
35     "LOAI": "Do an",  
36  
37     "VAT": 10  
38  
39   },  
40  
41   {  
42  
43     "MASO_SP": "SP03",  
44  
45     "TEN_SP": "Tiramisu",  
46  
47     "QUOCCGIA": "Anh",  
48  
49     "GIAGOC": 15000,  
50  
51     "GIABAN": 30000,  
52  
53     "LOAI": "Trang mieng",  
54  
55     "VAT": 10  
56  
57   },  
58  
59   {  
60  
61     "MASO_SP": "SP04",  
62  
63     "TEN_SP": "Lamington",  
64
```


-
- Nhập dữ liệu bảng QUANLYKHO_NHAP

Data Explorer

```

1 r.db('Jiangnam2').table('QUANLYKHO_NHAP').insert(
2
3 [
4
5   {
6
7     "MASO_CN": "CN2",
8
9     "MASO_SP": "SP01",
10
11     "NGAYNHAPKHO": "01/12/2023",
12
13     "SOLUONGNHAP": 88
14
15   },
16
17   {
18
19     "MASO_CN": "CN2",
20
21     "MASO_SP": "SP02",
22
23     "NGAYNHAPKHO": "01/12/2023",
24
25     "SOLUONGNHAP": 100
26
27   },
28
29   {
30
31     "MASO_CN": "CN2",
32
33     "MASO_SP": "SP04",
34
35     "NGAYNHAPKHO": "01/12/2023",
36
37     "SOLUONGNHAP": 198
38
39   },
40
41   {
42
43     "MASO_CN": "CN2",
44
45     "MASO_SP": "SP05",
46
47     "NGAYNHAPKHO": "01/12/2023",
48
49     "SOLUONGNHAP": 114
50
51   },
52
53   {
54
55     "MASO_CN": "CN2",
56
57     "MASO_SP": "SP08",
58
59     "NGAYNHAPKHO": "01/12/2023",
60
61     "SOLUONGNHAP": 10
62
63   },
64
65   {
66
67     "MASO_CN": "CN2",
68
69     "MASO_SP": "SP06",
70
71     "NGAYNHAPKHO": "01/12/2023",
72
73     "SOLUONGNHAP": 57
74
75   }
76 ]
77 )
78
79 );

```

- Nhập dữ liệu bảng QUANLYKHO_BAN

Data Explorer

```

1 r.db('Jiangnam2').table('QUANLYKHO_BAN').insert(
2
3 [
4
5   {
6     "MASO_CN": "CN2",
7     "MASO_SP": "SP01",
8     "TRANGTHAI": "Dang kinh doanh",
9     "SOLUONGTON": 18
10  },
11  {
12    "MASO_CN": "CN2",
13    "MASO_SP": "SP02",
14    "TRANGTHAI": "Dang kinh doanh",
15    "SOLUONGTON": 120
16  },
17  {
18    "MASO_CN": "CN2",
19    "MASO_SP": "SP03",
20    "TRANGTHAI": "Ngung nhap hang",
21    "SOLUONGTON": 20
22  },
23  {
24    "MASO_CN": "CN2",
25    "MASO_SP": "SP04",
26    "TRANGTHAI": "Dang kinh doanh",
27    "SOLUONGTON": 89
28  },
29  {
30    "MASO_CN": "CN2",
31    "MASO_SP": "SP05",
32    "TRANGTHAI": "Dang kinh doanh",
33    "SOLUONGTON": 205
34  },
35  {
36    "MASO_CN": "CN2",
37    "MASO_SP": "SP06",
38    "TRANGTHAI": "Dang nhap hang",
39    "SOLUONGTON": 114
40  },
41  {
42    "MASO_CN": "CN2",
43    "MASO_SP": "SP07",
44    "TRANGTHAI": "Ngung kinh doanh",
45    "SOLUONGTON": 0
46  }
47 ]
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86

```

- Nhập dữ liệu bảng HOADON

```
Data Explorer

1 r.db('Jiangnam2').table('HOADON').insert(
2
3 [
4
5 {
6
7   "MASO_HD": "HD06",
8
9   "MASO_NV": "NV05",
10
11  "MASO_KH": "KH10",
12
13  "MASO_CN": "CN2",
14
15  "NGAYXUAT": "07/11/2020",
16
17  "TONGTIEN": 200000
18
19 },
20
21 {
22
23  "MASO_HD": "HD07",
24
25  "MASO_NV": "NV06",
26
27  "MASO_KH": "KH01",
28
29  "MASO_CN": "CN2",
30
31  "NGAYXUAT": "02/01/2022",
32
33  "TONGTIEN": 3500000
34
35 },
36
37 {
38
39  "MASO_HD": "HD08",
40
41  "MASO_NV": "NV07",
42
43  "MASO_KH": "KH08",
44
45  "MASO_CN": "CN2",
46
47  "NGAYXUAT": "22/01/2024",
48
49  "TONGTIEN": 125000
50
51 },
52
53 {
54
55  "MASO_HD": "HD09",
56
57  "MASO_NV": "NV08",
58
59  "MASO_KH": "KH02",
60
61  "MASO_CN": "CN2",
62
63  "NGAYXUAT": "24/08/2023",
64
65  "TONGTIEN": 900000
66
67 },
68
69 {
70
71  "MASO_HD": "HD10",
72
73  "MASO_NV": "NV06",
74
75  "MASO_KH": "KH07",
76
77  "MASO_CN": "CN2",
78
79  "NGAYXUAT": "02/09/2021",
80
81  "TONGTIEN": 230000
82
83 },
84
85 {
```

- Nhập dữ liệu bảng CTHD

```
Data Explorer

1 r.db('Jiangnam2').table('CTHD').insert(
2
3
4 [
5   {
6     "MASO_HD": "HD06",
7     "MASO_SP": "SP01",
8     "SOLUONG": 3
9   },
10  {
11    "MASO_HD": "HD07",
12    "MASO_SP": "SP02",
13    "SOLUONG": 2
14  },
15  {
16    "MASO_HD": "HD08",
17    "MASO_SP": "SP05",
18    "SOLUONG": 2
19  },
20  {
21    "MASO_HD": "HD09",
22    "MASO_SP": "SP01",
23    "SOLUONG": 3
24  },
25  {
26    "MASO_HD": "HD10",
27    "MASO_SP": "SP02",
28    "SOLUONG": 2
29  },
30  {
31    "MASO_HD": "HD07",
32    "MASO_SP": "SP01",
33    "SOLUONG": 3
34  },
35  {
36    "MASO_HD": "HD09",
37    "MASO_SP": "SP02",
38    "SOLUONG": 2
39  },
40  {
41    "MASO_HD": "HD09",
42    "MASO_SP": "SP09",
43    "SOLUONG": 2
44  },
45  ],
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

2. Truy vấn giữa hai máy

2.1. Máy 1 thực hiện

a) Xem tất cả nhân viên ở cả hai chi nhánh.

Data Explorer History Settings Fullscreen

```
1 r.db('Jiangnam1').table('NHANVIEN').union(  
2 r.db('Jiangnam2').table('NHANVIEN')  
3 )
```

Clear Run

8 rows returned. Displaying rows 0-7

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	GIOITINH	LUONG	MASO_CN	MASO_NV	TEN_NV
0	Nam	3000000	CN1	NV03	Dang Nhat Nam
1	Nu	6000000	CN1	NV04	Bui My Linh
2	Nam	4050000	CN1	NV02	Tran Anh Khoi
3	Nu	5000000	CN1	NV01	Le Thi Hanh
4	Nam	5500000	CN2	NV07	Thai Minh Tri
5	Nam	6000000	CN2	NV08	Dao Cong Tuan
6	Nu	4500000	CN2	NV06	Phan Bich Ngoc
7	Nu	7500000	CN2	NV05	Nguyen Minh Thu

b) Thống kê tổng số tiền mỗi khách hàng đã thanh toán ở hai chi nhánh.

Data Explorer History Settings Fullscreen

```
1 r.db('Jiangnam1').table('KHACHHANG').merge(function(khachhang){  
2 return {  
3 TONGTIENTHANHTOAN: r.db('Jiangnam1').table('HOADON')  
4 .union(r.db('Jiangnam2').table('HOADON')).filter({MASO_KH: khachhang('MASO_KH')}).sum('TONGTIEN')  
5 }  
6 })  
7 .pluck('MASO_KH', 'TEN_KH', 'TONGTIENTHANHTOAN')
```

Clear Run

12 rows returned. Displaying rows 0-11

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	MASO_KH	TEN_KH	TONGTIENTHANHTOAN
0	KH07	Nguyen Thao Ly	423500
1	KH11	Vo Diem My	0
2	KH05	Ngo Chan Huy	0
3	KH03	Nguyen Phuc Hao	23000
4	KH06	Nguyen Phuong Nhi	0
5	KH01	Nguyen Anh Tai	8992500
6	KH04	Nguyen Xuan Thi	0
7	KH02	Chu Huyen Trang	2248500
8	KH09	Phan Phuc Hao	0
9	KH12	Nguyen Phuong Hang	0
10	KH08	Pham Nguyen Lam	125000
11	KH10	Kim Si Vuong	200000

c) Cập nhật giá bán tăng 10% với tất cả sản phẩm thuộc loại ‘Trang mieng’ ở hai chi nhánh.

Data Explorer History Settings

```

1 r.db('Jiangnam1').table('SANPHAM')
2 .filter(r.row('LOAI').eq('Trang mieng'))
3 .update({
4   GIABAN: r.row('GIABAN').mul(1.1)
5 });
6
7 r.db('Jiangnam2').table('SANPHAM')
8 .filter(r.row('LOAI').eq('Trang mieng'))
9 .update({
10  GIABAN: r.row('GIABAN').mul(1.1)
11 });

```

Clear Run

1 row returned in 25ms.

Tree view Table Viewer View Table view Raw view Query profile

	deleted	errors	inserted	replaced	skipped	unchanged
0	0	0	0	3	0	0

d) Cập nhật sản phẩm ‘Dang nhap hang’ với những sản phẩm có mã ‘SP03’ ở chi nhánh 2.

Data Explorer History Settings

```

1 r.db('Jiangnam2').table('QUANLYKHO_BAN')
2 .filter({MASO_SP: "SP03"})
3 .update({TRANGTHAI: "Dang nhap hang"})

```

Clear Run

1 row returned in 6ms.

Tree view Table Viewer View Table view Raw view Query profile

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

e) Xóa những sản phẩm có trạng thái là ‘Ngung kinh doanh’ ở chi nhánh 2.

Data Explorer History Settings

```

1 r.db('Jiangnam2').table('QUANLYKHO_BAN')
2 .filter({TRANGTHAI: "Ngung kinh doanh"})
3 .delete()

```

Clear Run

1 row returned in 9ms.

Tree view Table Viewer View Table view Raw view Query profile

```

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

```

2.2. Máy 2 thực hiện

a) Xem tình những sản phẩm có số lượng tồn hơn 50 ở cả hai chi nhánh.

Data Explorer History Settings

```
1 r.db('Jiangnam1').table('QUANLYKHO_BAN').eqJoin(  
2 'MASO_SP',  
3 r.db('Jiangnam1').table('SANPHAM')  
4 ).zip()  
5 .filter(function(sp) {return sp('SOLUONGTON').gt(50) })  
6 .union(  
7 r.db('Jiangnam2').table('QUANLYKHO_BAN').eqJoin(  
8 'MASO_SP',  
9 r.db('Jiangnam2').table('SANPHAM')  
10 ).zip()  
11 .filter(function(sp) {return sp('SOLUONGTON').gt(50) })  
12 )  
13 .pluck('MASO_CN', 'MASO_SP', 'TEN_SP', 'SOLUONGTON')
```

Clear Run

11 rows returned. Displaying rows 0-10

Tree view Table Viewer View Table view Raw view Query profile

	MASO_CN	MASO_SP	SOLUONGTON	TEN_SP
0	CN1	SP10	150	Nuoc dau tam
1	CN1	SP06	114	Gimbap
2	CN1	SP01	88	Oreo
3	CN1	SP04	198	Lamington
4	CN1	SP03	123	Tiramisu
5	CN2	SP10	100	Nuoc dau tam
6	CN2	SP06	114	Gimbap
7	CN2	SP04	89	Lamington
8	CN2	SP02	120	Kitkat
9	CN2	SP08	80	Croissant
10	CN2	SP05	205	Tobboki

b) Thống kê những khách hàng chưa từng mua hàng ở cả hai chi nhánh.

Data Explorer History Settings

```
1 r.db('Jiangnam1').table('KHACHHANG').merge(function(khachhang){  
2 return {  
3 TONGTIENHANHTOAN: r.db('Jiangnam1').table('HOADON')  
4 .union(r.db('Jiangnam2').table('HOADON')).filter({MASO_KH: khachhang('MASO_KH')}).sum('TONGTIEN')}}  
5 ).filter({TONGTIENHANHTOAN : 0})  
6 .pluck('MASO_KH', 'TEN_KH', 'TONGTIENHANHTOAN')
```

Clear Run

6 rows returned. Displaying rows 0-5

Tree view Table Viewer View Table view Raw view Query profile

	MASO_KH	TEN_KH	TONGTIENHANHTOAN
0	KH11	Vo Diem My	0
1	KH05	Ngo Chan Huy	0
2	KH12	Nguyen Phuong Hang	0
3	KH06	Nguyen Phuong Nhi	0
4	KH04	Nguyen Xuan Thi	0
5	KH09	Phan Phuc Hao	0

c) Thống kê nhân viên bán được nhiều đơn nhất ở cả 2 chi nhánh.

Data Explorer History Settings

```
1 r.db('Jiangnam1').table('HOADON')
2 .group('MASO_NV')
3 .count()
4 .ungroup()
5 .map(function (nv) {return{
6   MASO_CN: r.db('Jiangnam1').table('NHANVIEN').get(nv('group'))('MASO_CN'),
7   MASO_NV: nv('group'),
8   TEN_NV: r.db('Jiangnam1').table('NHANVIEN').get(nv('group'))('TEN_NV'),
9   SO_DON: nv('reduction')
10 }})
11 .orderBy(r.desc('SO_DON'))
12 .limit(1)
13 .union(
14   r.db('Jiangnam2').table('HOADON')
15   .group('MASO_NV')
16   .count()
17   .ungroup()
18   .map(function (nv) {return{
19     MASO_CN: r.db('Jiangnam2').table('NHANVIEN').get(nv('group'))('MASO_CN'),
20     MASO_NV: nv('group'),
21     TEN_NV: r.db('Jiangnam2').table('NHANVIEN').get(nv('group'))('TEN_NV'),
22     SO_DON: nv('reduction')
23   }})
24   .orderBy(r.desc('SO_DON'))
25   .limit(1)
26 )
27
```

Clear Run

2 rows returned in 19ms.

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

	MASO_CN	MASO_NV	SO_DON	TEN_NV
0	CN1	NV01	3	Le Thị Hanh
1	CN2	NV05	2	Nguyen Minh Thu

d) Cập nhật tăng lương cho nhân viên có mã nhân viên là 'NV03' ở chi nhánh 1.

Data Explorer History Settings

```
1 r.db('Jiangnam1').table('NHANVIEN')
2 .filter({MASO_CN: 'CN1', MASO_NV: 'NV03'})
3 .update({LUONG: r.row('LUONG').mul(1.1)})
```

Clear Run

1 row returned in 8ms.

[Tree view](#) [Table Viewer View](#) [Table view](#) [Raw view](#) [Query profile](#)

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


e) Xóa khách hàng có mã là ‘KH01’ ở chi nhánh 1.

Data Explorer History Settings Fullscreen

```
1 r.db('Jiangnam1').table('KHACHHANG')
2 .filter({MASO_KH: "KH01"})
3 .delete()
```

Clear Run

1 row returned in 6ms.

Tree view Table Viewer View Table view Raw view Query profile

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

3. CƠ CHẾ NHÂN BẢN TRONG PHÂN TÁN TRONG RETHINKDB

“Sharding and replication”: Đây là thuật ngữ dùng trong quản lý cơ sở dữ liệu để chỉ việc phân chia dữ liệu thành nhiều phần (shards) để phân tải và tăng hiệu suất, cũng như sao chép dữ liệu để đảm bảo tính sẵn sàng và dự phòng.

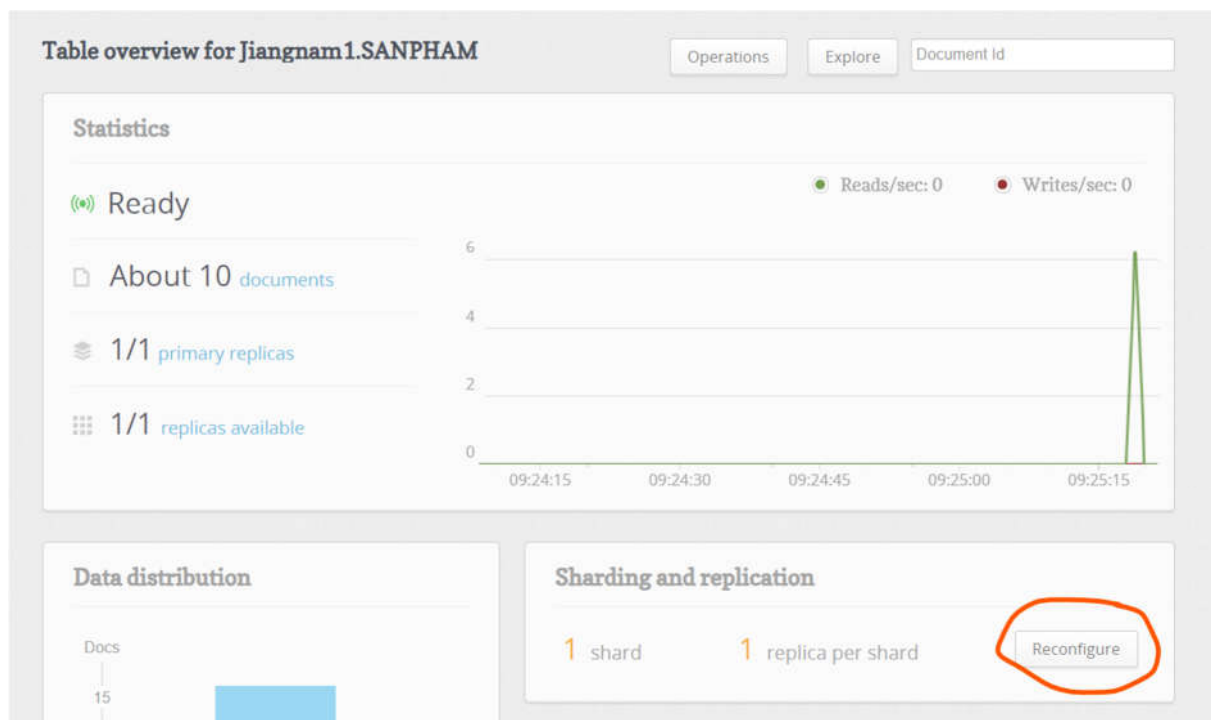
Nhân bản là việc thực hiện copy dữ liệu từ server này sang server khác. Mục đích của việc nhân bản là để dự phòng và tăng tính khả dụng của dữ liệu.

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.

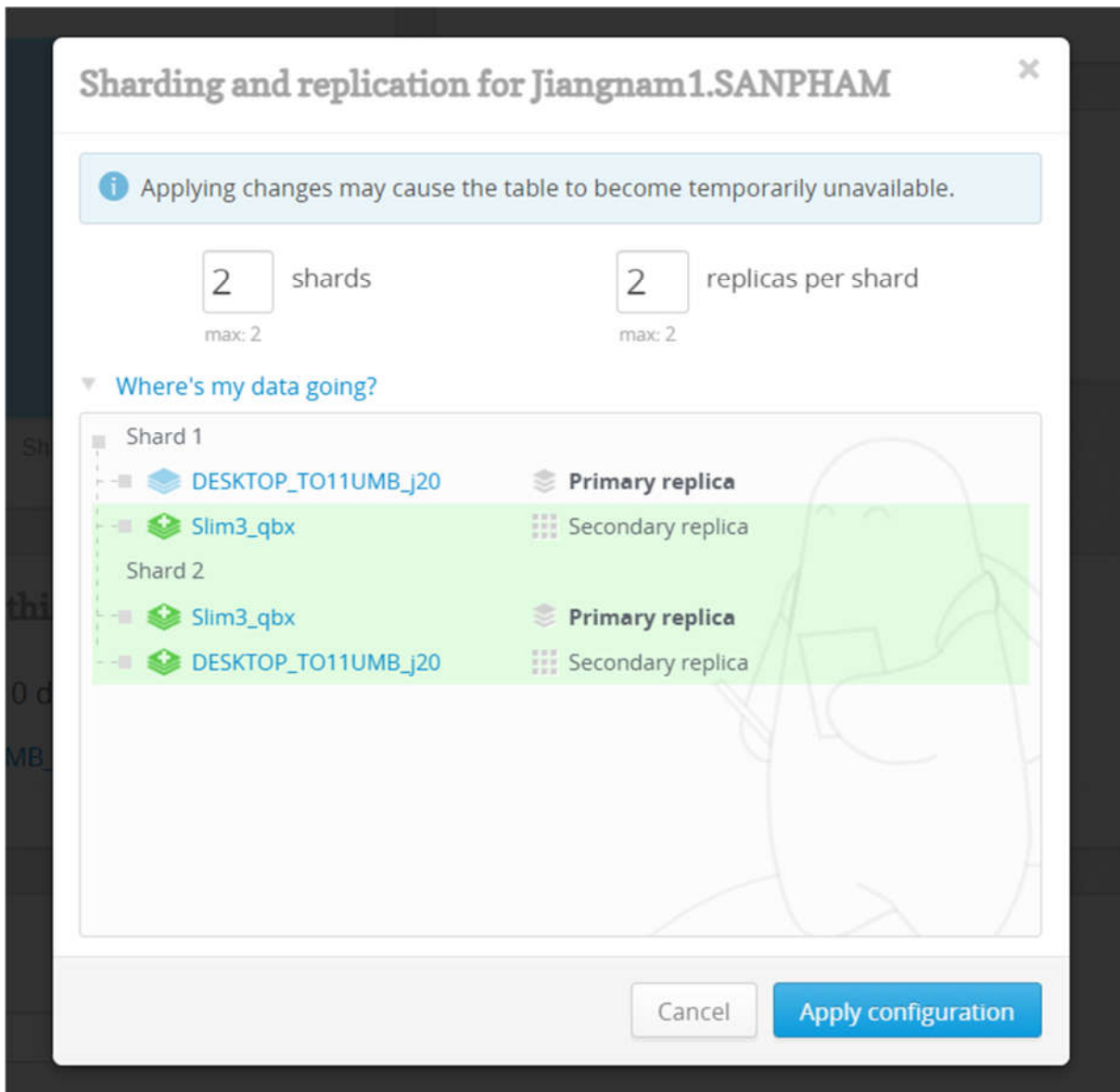
Cách nhân bản 1 table trên RethinkDB:

- **Bước 1: Chọn bảng cần được nhân bản.**

Ở đây nhóm chọn bảng SANPHAM

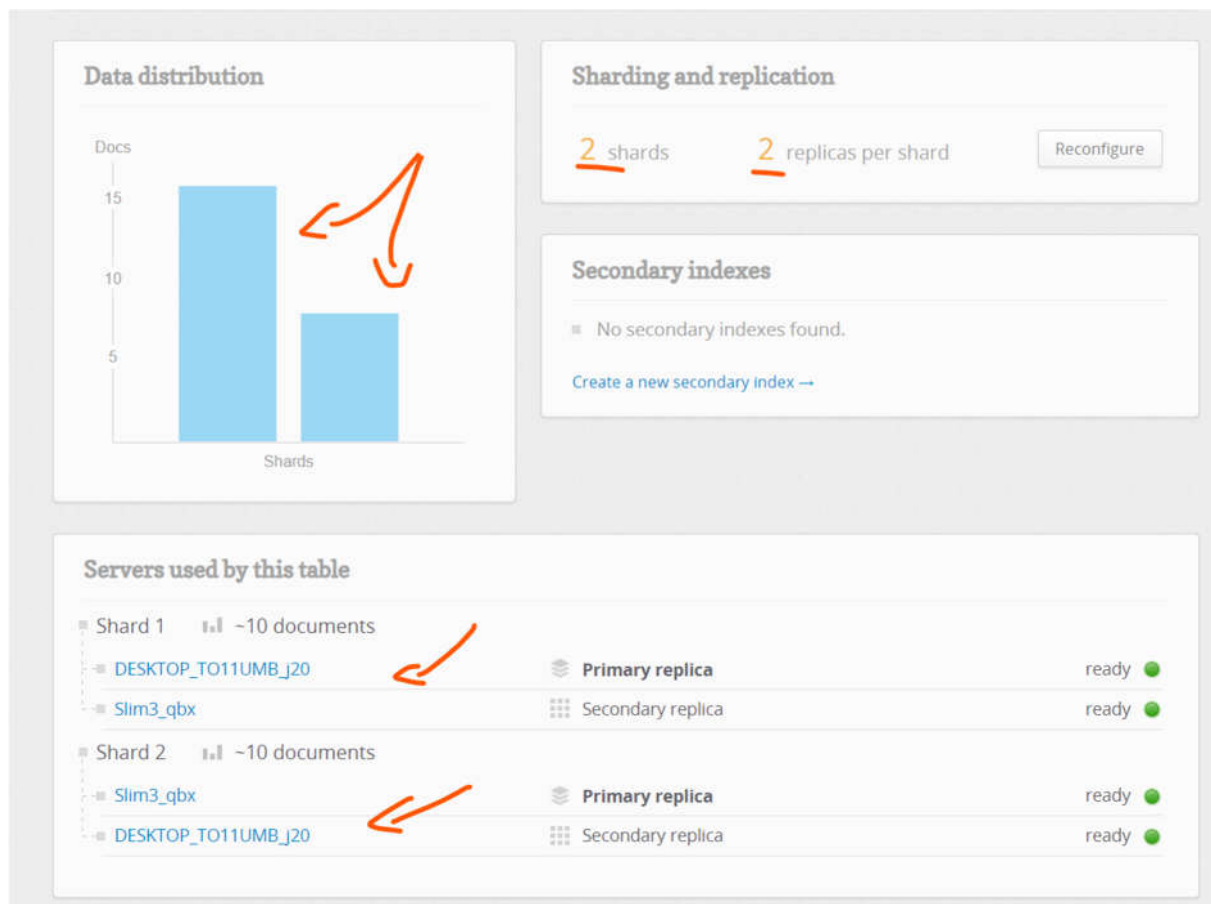


- **Bước 2: Chọn Reconfigure**



- **Bước 3: Chọn số lượng shard (phân mảnh) và chọn số lượng replicas (nhân bản) cho mỗi shard sau đó nhấn Apply.**

Sau khi nhân bản xong ta có thể thấy số lượng replicas là 2 trên mỗi mảnh (shard)



CHƯƠNG 4: TÀI LIỆU THAM KHẢO

1. <https://codelearn.io/sharing/ban-biet-gi-ve-nosql-database>
2. <https://viblo.asia/p/gioi-thieu-ve-he-quan-tri-co-so-du-lieu-rethinkdb-jvEla4O6Zkw>
3. <https://rethinkdb.com/faq>
4. <https://rethinkdb.com/api/javascript/>
5. <https://rethinkdb.com/docs/introduction-to-reql/>