**Shape

Description automatically generated**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**Đồ Án**

Môn học: Quản Lý Thông Tin

***Lớp****:*

IE103.O11

***Sinh viên thực hiện:***

Nguyễn Lê Tấn Quang– 21522512

Phạm Anh Kiệt – 21522260

Hoắc Công Minh – 21522334

***Giảng viên hướng dẫn:***

TS. Nguyễn Gia Tuấn Anh

CN. Trần Quốc Khánh

**TP. HỒ CHÍ MINH – 12/2023**

**TÓM TẮT BÁO CÁO**

**1. Tiêu đề báo cáo: TÌM HIỂU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NEO4J**

**2. Bảng phân công công việc:**

|  |  |  |
| --- | --- | --- |
| MSSV | Họ tên | Công việc |
| 21522512 | Nguyễn Lê Tấn Quang |  |
| 21522260 | Phạm Anh Kiệt |  |
| 21522334 | Hoắc Công Minh |  |

**3. Nội dung báo cáo:**

**Nội dung 1: Giới thiệu sơ lược về hệ quản trị cơ sở dữ liệu Neo4j**

+ Đề mục 1: Đặc điểm chính của Neo4j

+ Đề mục 2: Nhà sáng lập và lịch sử phát triển

+ Đề mục 3: Mã nguồn và mô hình hỗ trợ

+ Đề mục 4: Ứng dụng của Neo4j

**Nội dung 2: Một số khái niệm cơ bản trong Neo4j**

+ Đề mục 1: Cơ sở dữ liệu đồ thị là gì?

+ Đề mục 2: Node

+ Đề mục 3: Relationship

+ Đề mục 4: Ngôn ngữ truy vấn Cypher

**Nội dung 3: Các tính năng cơ bản**

+ Đề mục 1: Đọc dữ liệu

+ Đề mục 2: Ghi Dữ liệu

+ Đề mục 3: Tạo Node

+ Đề mục 4: Cập nhật Node

+ Đề mục 5: Xóa Node

+ Đề mục 6: Các thao tác với Relationship

**Nội dung 4: Các tính năng nâng cao**

+ Đề mục 1: Phân quyền

+ Đề mục 2: Xác thực

+ Đề mục 3: Backup và Restore

+ Đề mục 4: Import và Export

**Nội dung 5: So sánh với một số hệ quản trị cơ sở dữ liệu khác**

**Nội dung 6: Kết luận và hướng phát triển**

+ Đề mục 1: Kết luận

+ Đề mục 2: Hướng phát triển

Mục Lục

[Mục Lục 2](#_Toc154360080)

[Danh mục hình ảnh 3](#_Toc154360081)

[Danh mục Bảng 4](#_Toc154360082)

[LỜI CẢM ƠN 5](#_Toc154360083)

[NHẬN XÉT 6](#_Toc154360084)

[1.1. Đặc điểm nổi bật của RethinkDB 6](#_Toc154360085)

[1.2. So sánh với các HQT CSDL khác 7](#_Toc154360086)

[CHƯƠNG 1: TỔNG QUAN VỀ RETHINKDB 8](#_Toc154360087)

[1.1. Sơ lược về hệ quản trị RethinkDB 8](#_Toc154360088)

[1.2. Lịch sử phát triển 8](#_Toc154360089)

[1.3. Cộng đồng và sự hỗ trợ 9](#_Toc154360090)

[1.4. Mô hình dữ liệu và mô hình lưu trữ 10](#_Toc154360091)

[1.5. Ngôn ngữ truy vấn và thao tác dữ liệu 12](#_Toc154360092)

[1.6. Cơ chế phân tán 14](#_Toc154360093)

[CHƯƠNG 2: CÀI ĐẶT VÀ CẤU HÌNH 17](#_Toc154360094)

[2.1 Cài đặt Rethinkdb trên một máy 17](#_Toc154360095)

[2.2 Cài đặt Rethinkdb cho cụm máy phân tán 22](#_Toc154360096)

[CHƯƠNG 3: THỰC NGHIỆM MÔ PHỎNG PHÂN TÁN 27](#_Toc154360097)

[3.1 Mô tả bài toán đặt ra với dữ liệu 27](#_Toc154360098)

[3.2 Mô tả cấu trúc dữ liệu sử dụng 27](#_Toc154360099)

[3.3 Thực nghiệm phân tán 28](#_Toc154360100)

[Summary 34](#_Toc154360101)

[Tài liệu tham khảo 35](#_Toc154360102)

# Danh mục hình ảnh

[Hình 42: Xem dữ liệu vừa thêm bên cả 2 máy 33](#_Toc154318082)

# Danh mục Bảng

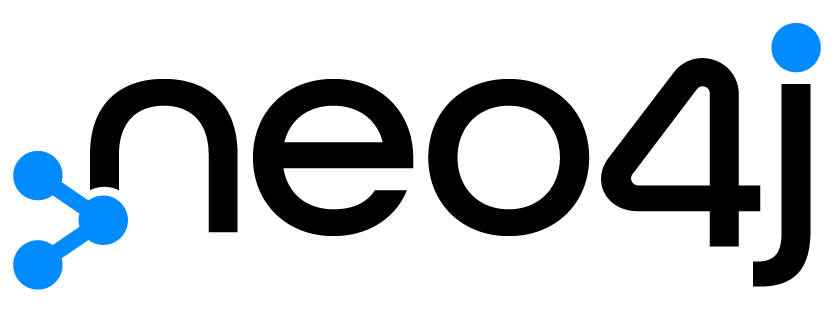
[Bảng 1: So sánh với các hệ quản trị 7](#_Toc154317658)

# CHƯƠNG 1: GIỚI THIỆU SƠ LƯỢC VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU NEO4J

## Đặc điểm chính:

Neo4j là một hệ quản trị cơ sở dữ liệu đồ thị (Graph Database Management System – GDBMS), nổi bật với khả năng lưu trữ và quản lý dữ liệu theo dạng đồ thị.

Cung cấp khả năng truy xuất dữ liệu hiệu quả thông qua cấu trúc đồ thị, phù hợp cho các ứng dụng có mối quan hệ phức tạp giữa các đối tượng.



Hình 1. Logo của Neo4j

## Nhà sáng lập và lịch sử phát triển

RethinkDB được thành lập vào năm 2009 bởi Slava Akhmechet và Michael Glukhovsky. Phiên bản đầu tiên của RethinkDB là một công cụ lưu trữ được tối ưu hóa cho SSD cho MySQL. Sau đó họ chuyển sang xây dựng một hệ quản trị cơ sở dữ liệu hướng tài liệu giống như MongoDB.

Bản phát hành đầu tiên của kiến trúc RethinkDB DBMS hiện tại bắt đầu vào tháng 11 năm 2012. Phiên bản đầu tiên này hỗ trợ mô hình dữ liệu JSON, cũng như tính nhất quán tức thời, bản đồ/rút gọn kiểu Hadoop, chia sẻ, sao chép đa trung tâm dữ liệu và chuyển đổi dự phòng. Vào tháng 6 năm 2013, RethinkDB đã giới thiệu các tính năng mới cho ReQL, chẳng hạn như kiểm soát truy cập cơ bản, kết hợp biểu thức chính quy, thao tác mảng...

Phiên bản 2.0 của RethinkDB năm 2012 là phiên bản "prodcution ready" đầu tiên. Vào tháng 8 năm 2015, nó đã hỗ trợ chuyển đổi dự phòng tự động bằng giao thức dựa Raft. Vào tháng 11 năm 2015, RethinkDB đã giới thiệu tính năng nổi bật là Changefeeds giúp dữ liệu đảm bảo được cập nhật liên tục.

Vào tháng 10 năm 2016, công ty thành lập RethinkDB đóng cửa. Lý do là họ không thể giữ được sự bền vững trong quá trình xây dựng công ty. Sau một năm, mã nguồn đã được Cloud Native Computing Foundation (CNCF) mua lại, sau đó nó được phát hành trở lại nỗ lực của cộng đồng nguồn mở vào tháng 7 năm 2017. CNCF là là một dự án của Linux Foundation được thành lập vào năm 2015, tổ chức này có chức năng chính là hỗ trợ phát triển các công nghệ lưu trữ tiến bộ và các ứng dụng hệ thống xung quanh các công nghệ này. [2]

## 1.3. Cộng đồng và sự hỗ trợ

Cộng đồng và sự hỗ trợ đóng một vai trò quan trọng trong sự thành công của bất kỳ dự án phần mềm nào, bao gồm cả RethinkDB. Hệ quản trị này cung cấp một dãy những nền tảng cho người dùng trao đổi, học tập, cũng như phát triển mã nguồn mở. Có thể kể đến một số kênh như Slack Commnunity, với hơn 7000 cộng tác viên. Ngoài ra còn có những kênh trao đổi thông tin như Twitter, Discord hoặc rộng hơn người dùng có thể tìm kiếm giải đáp cho thắc mắc của mình qua StackOverflow. Một nguồn nữa cũng không kém phần quan trọng là GitHub repo, nơi để lập trình viên có thể báo cáo những lỗi phát hiện, gợi ý một số tính năng mới cho hệ quản trị cũng như theo dõi tiến độ phát triển của mã nguồn. [2]

Ngoài ra, còn có các nhà tài trợ RethinkDB ngoài CNCF. Đầu tiên có thể nhắc đến:

Digital Ocean: được thành lập vào năm 2012 để đáp ứng nhu cầu của các nhà phát triển đang tìm kiếm giải pháp điện toán đám mây đơn giản, giá cả phải chăng. Nhà tài trợ này cung cấp cơ sở hạ tầng và máy chủ cần thiết để phục vụ các trang web quan trọng như download.rethinkdb.com hoặc update.rethinkdb.com [3]

Atlassian: Atlassian Corporation Plc là một công ty phần mềm có trụ sở tại Vương quốc Anh, nhưng có nguồn gốc từ Úc và Hoa Kỳ. Công ty chuyên phát triển sản phẩm cho các lập trình viên phần mềm, quản lý dự án và các nhóm phát triển phần mềm khác. Atlassian cung cấp giấy phép OSS để có thể xử lý các yêu cầu nội bộ như các vấn đề về lỗ hổng. [3]

Netlify: là công ty điện toán đám mây đầu tiên từ xa cung cấp nền tảng phát triển bao gồm các dịch vụ xây dựng, triển khai và phụ trợ không cần máy chủ cho các ứng dụng web và trang web động. Netlify cung cấp giấy phép OSS để rethinkdb.com có thể hoạt động. [3]

DNsimple: cung cấp dịch vụ DNS cho dự án RethinkDB [3]

ZeroTier: tài trợ cho việc phát triển tính năng ghi tổng hợp có thể cấu hình cho từng bảng, bao gồm khả năng thiết lập độ trễ ghi là vô cùng để tạo ra một bảng chỉ tồn tại trong bộ nhớ (PR #6392). [3]

## 1.4. Mô hình dữ liệu và mô hình lưu trữ

- Mô hình lưu trữ: RethinkDB là một cơ sở dữ liệu hướng tài liệu(document-oriented), RethinkDB được xây dựng để lưu trữ các tài liệu JSON và mở rộng quy mô sang nhiều máy một cách dễ dàng. RethinkDB có một ngôn ngữ truy vấn hỗ trợ các truy vấn như table joins and group by. [4]

- Mô hình dữ liệu: Mô hình dữ liệu RethinkDB (data model) bao gồm hai thành phần chính: kiểu dữ liệu trong RethinkDB và mô hình quan hệ trong Rethink DB. [5]

A diagram of data model

Description automatically generated

Hình 1: Sơ đồ về mô hình dữ liệu RethinkDB

- Kiểu dữ liệu (RethinkDB data types): Các kiểu dữ liệu cơ bản của Rethink DB hỗ trợ

* Numbers (Số): RethinkDB lưu trữ các số bằng cách sử dụng số dấu phẩy động với độ chính xác cao. [5]
* Strings (Chuỗi): Chuỗi được lưu trữ theo tiêu chuẩn UTF-8, cho phép xử lý và lưu trữ các ký tự đa ngôn ngữ.
* Objects (Đối Tượng): Được lưu trữ dưới dạng cặp khóa-giá trị theo định dạng JSON tiêu chuẩn. Điều này cung cấp cấu trúc dữ liệu phong phú và linh hoạt.
* Arrays (Mảng): Lưu trữ dưới dạng danh sách các phần tử và hỗ trợ đến 100.000 phần tử theo mặc định.
* Binary (Nhị Phân): Các đối tượng nhị phân bao gồm các tệp, hình ảnh và dữ liệu nhị phân khác.
* Date and Time (Ngày và Giờ): RethinkDB lưu trữ thông tin ngày và giờ với độ chính xác đến mili giây, hỗ trợ xử lý thời gian một cách chính xác.
* Booleans (Luận Lý): Giá trị đúng và sai, sử dụng để biểu diễn các điều kiện luận lý.
* Null: Lưu trữ giá trị null, thường được sử dụng để biểu diễn sự không xác định.

- Mô hình quan hệ trong RethinkDB (RethinkDB model relationships):

RethinkDB, ngoài việc là một cơ sở dữ liệu NoSQL, cung cấp một trong những tính năng được yêu cầu nhất bởi các nhà phát triển SQL, đó là khả năng thực hiện các truy vấn JOIN. Có hai cách để mô hình hóa mối quan hệ trong RethinkDB. [5]

* Sử dụng mảng nhúng trong tài liệu (Embedded arrays in the document): sử dụng mảng nhúng là sử dụng một mảng các đối tượng trong bảng cho mỗi tài liệu.
* Liên kết các tài liệu được lưu trữ trong nhiều bảng (Linking documents stored in multiple tables): tương tự như mối quan hệ khóa trong cơ sở dữ liệu SQL truyền thống.

## 1.5. Ngôn ngữ truy vấn và thao tác dữ liệu

**- Ngôn ngữ truy vấn**

ReQL (RethinkDB Query Language) là ngôn ngữ truy vấn của hệ quản trị cơ sở dữ liệu RethinkDB. ReQL là một ngôn ngữ truy vấn tùy chỉnh được thiết kế để mạnh mẽ và dễ hiểu. ReQL cung cấp một cú pháp rõ ràng có thể được sử dụng để thao tác trên các tài liệu JSON trong cơ sở dữ liệu. Nó được xây dựng trên ba nguyên tắc chính: [6]

* Nhúng vào Ngôn Ngữ Lập Trình của Bạn (ReQL embeds into your programming language): ReQL được tích hợp trực tiếp vào ngôn ngữ lập trình đang sử dụng. Các truy vấn được tạo ra bằng cách gọi các hàm trong ngôn ngữ lập trình quen thuộc. Không cần phải làm việc với việc nối chuỗi hoặc tạo các đối tượng JSON đặc biệt để truy vấn cơ sở dữ liệu.
* Tất cả các truy vấn ReQL đều có thể kết nối được (All ReQL queries are chainable): Có thể bắt đầu với một bảng và sau đó liên kết các biến dần dần cho đến cuối truy vấn, sử dụng toán tử "." (chấm).
* Thực Hiện Tất Cả Truy Vấn Trên Máy Chủ(All queries execute on the server): Tất cả các truy vấn ReQL đều được thực hiện trên máy chủ. Mặc dù tạo ra truy vấn trên máy khách bằng ngôn ngữ lập trình quen thuộc, nhưng chúng được thực hiện toàn bộ trên máy chủ cơ sở dữ liệu khi gọi lệnh "run" và truyền nó một kết nối cơ sở dữ liệu đang hoạt động.

**- Thao tác dữ liệu**

Dưới đây là bảng so sánh thuật ngữ tương đương giữa SQL và RethinkDB [7]:

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

Bảng : So sánh thuật ngữ tương đương giữa SQL và RethinkDB

Dựa vào bảng trên có thể thấy sự khác nhau:

* Tài liệu (document) trong RethinkDB tương đương với hàng (row) trong SQL
* Trường (field) trong RethinkDB tương đương với cột (column) trong SQL
* Khóa chính mặc định trong RethinkDB là id

- Các câu lệnh thao tác dữ liệu cơ bản [7]:

| **Thao tác** | **SQL** | **RethinkDB** |
| --- | --- | --- |
| Tạo Database | CREATE DATABASE <Tên database> | r.db\_create(‘Tên db’).run() |
| Tạo bảng | CREATE TABLE <T ên table> {  Thuộc tính 1,  Thuộc tính 2,  Thuộc tính n,..  } | r.table\_create(‘Tên table’).run() |
| Chèn dữ liệu | INSERT INTO table\_name (column1, column2) VALUES (value1, value2) | r.table('table\_name').insert({field1: value1, field2: value2}) |
| Truy vấn dữ liệu | SELECT column1, column2 FROM table\_name | r.table(“table\_name”).pluck(“field1”, “field2”) |
| Lọc dữ liệu | SELECT \* FROM table WHERE condition | r.table('table\_name').filter(condition) |
| Sắp xếp | SELECT \* FROM table\_name ORDER BY column ASC/DESC | r.table('table\_name').orderBy(r.asc/desc('field')) |
| Cập nhật | UPDATE table\_name SET column = value WHERE condition | r.table('table\_name').filter(condition).update({field: value}) |
| Xóa | DELETE FROM table\_name WHERE condition | r.table('table\_name').filter(condition).delete() |
| Join bảng | SELECT \* FROM t1 JOIN t2 ON t1.id = t2.t1\_id | r.table('t1').eqJoin('id', r.table('t2')).zip() |

Bảng : So sánh các lệnh thao tác dữ liệu giữa SQL và RethinkDB

## 1.6. Cơ chế phân tán

RethinkDB được thiết kế với kiến trúc phân tán để đáp ứng yêu cầu của các ứng dụng có thể mở rộng và đảm bảo tính sẵn sàng cao.

* **Sharding (Phân mảnh) và Replication (Sao chép)**

RethinkDB phân vùng dữ liệu thành các phân đoạn bằng thuật toán phân chia phạm vi range sharding algorithm) dựa trên khóa chính của bảng . Khi người dùng chỉ định số lượng phân đoạn cho một bảng, RethinkDB sẽ phân tích số liệu thống kê của bảng và xác định các điểm phân chia tối ưu để phân bổ đều dữ liệu trên các phân đoạn. Sharding hiện dựa trên khóa chính của bảng và không thể thực hiện dựa trên các thuộc tính khác. Hệ thống chọn các điểm phân chia để đảm bảo rằng mỗi phân đoạn chứa số lượng tài liệu tương tự nhau, ngay cả khi các khóa chính được phân bổ không đồng đều. Điểm phân chia không được tự động thay đổi sau khi tạo bảng, nhưng người dùng có thể cân bằng lại các phân đoạn theo cách thủ công khi cần.

RethinkDB hỗ trợ sao chép (replication) để đảm bảo tính sẵn sàng và an toàn dữ liệu. Các bản sao của dữ liệu (replicas) được tạo ra và phân tán trên các máy chủ khác nhau, tăng khả năng chịu lỗi và khả năng phục hồi sau sự cố.

Cấu hình shard(phân mảnh) và sao chép(replication) được thực hiện thông qua cấu hình bảng, cho phép bạn đơn giản chỉ định số lượng shard và bản sao cho mỗi bảng hoặc tất cả các bảng trong một cơ sở dữ liệu. Người dùng không cần phải liên kết máy chủ với bảng một cách thủ công. RethinkDB sử dụng một tập hợp các phương pháp phỏng đoán để cố gắng đáp ứng các cấu hình bảng theo cách tối ưu. Nó sẽ sao chép dữ liệu cho các bản sao mới từ một máy chủ có sẵn, phân phối đồng đều các bản sao dữ liệu trên toàn cụm, cố gắng phân phối tải đồng đều. [8]

* **Định lí CAP**:

Định lí CAP là một nguyên tắc trong hệ thống phân tán, nói về việc một hệ thống không thể đồng thời đảm bảo Tính nhất quán (Consistency), Khả Dụng (Availability), và Dung sai phân vùng (Partition Tolerance) trong mọi tình huống. [9]

RethinkDB, giống như nhiều hệ thống cơ sở dữ liệu phân tán khác, phải đối mặt với Định lý CAP. Hệ thống không thể đảm bảo đồng thời tính nhất quán (Consistency), khả dụng (Availability), và dung sai phân vùng (Partition Tolerance) trong mọi tình huống. Trong đó, RethinkDB chủ yếu tập trung vào tính nhất quán và khả dụng.

* Đồng Nhất (Consistency): RethinkDB được thiết kế để duy trì tính đồng nhất ngay lập tức (immediately consistent). Mỗi shard trong RethinkDB được gán cho một bản sao chính quyền duy nhất (primary replica). Mọi đọc và ghi đều được định tuyến đến primary tương ứng, nơi chúng được sắp xếp và đánh giá. Đọc theo sau một ghi được công nhận luôn đảm bảo thấy được ghi đó. [8]
* Khả Dụng (Availability): RethinkDB hỗ trợ cả đọc theo cách hiện tại (up-to-date) và đọc không hiện tại (out-of-date). Theo mặc định, mọi truy vấn đọc được thực hiện theo cách hiện tại, tức là đều được định tuyến đến primary và thực hiện theo thứ tự với các hoạt động khác trên shard. Trong chế độ mặc định này, client luôn thấy phiên bản mới nhất, đồng nhất và không lỗi của dữ liệu. [8]

# CHƯƠNG 2: CÀI ĐẶT VÀ CẤU HÌNH

## 2.1 Cài đặt Rethinkdb trên một máy

**Bước 1: Tải về và cài đặt.**

Để có thể tải và cài đặt đúng rethinkdb, chúng ta có thể tham khảo documents của db tại: <https://rethinkdb.com/docs/start-on-startup/?fbclid=IwAR1KmYzCAlMtbAw8gjP32peMvxp1PVnZ9b4Gp-BilYjUhTw5VzEvgcsCbEE.>

**1a. Tải về public key bằng câu lệnh sau:**

**wget -qO- https://download.rethinkdb.com/repository/raw/pubkey.gpg | \**

**sudo gpg --dearmor -o /usr/share/keyrings/rethinkdb-archive-keyrings.gpg**

A screenshot of a computer program

Description automatically generated

Hình 2: Tải Rethinkdb

**1b. Thêm nguồn cài đặt của Rethinkdb vào kho lưu trữ(repository) của Ubuntu bằng câu lệnh sau:**

**echo "deb [signed-by=/usr/share/keyrings/rethinkdb-archive-keyrings.gpg]**

**https://download.rethinkdb.com/repository/ubuntu-$(lsb\_release -cs) $ (lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/rethinkdb.list**

A screen shot of a computer program

Description automatically generated

Hình 3: Thêm vào repository

**1c. Cập nhật danh sách các gói phần mềm bao gồm thông tin và các gói cơ sở dữ liệu, qua câu lệnh:**

**sudo apt-get update**

A screenshot of a computer screen

Description automatically generated

Hình 4: Cập nhật gói rethinkdb

**1d. Cài đặt môi trường**

**sudo apt-get install rethinkdb**

A screenshot of a computer

Description automatically generated

Hình 5: Cài đặt

**1e. Kiểm tra phiên bản Rethinkdb**

**rethinkdb --vesion**



Hình 6: Kiểm tra phiên bản

**Bước 2: Sử dụng Vim để tạo file chứa server.**

File này có thể tuỳ chỉnh tên server và dùng để cài đặt hệ thống phân tán, file có đuôi **“.conf”**

File nên được đặt tại mục: **/etc/rethinkdb/instances.d**

Câu lệnh đầy đủ cho hành động này:

**sudo vi /etc/rethinkdb/instances.d/<tên\_file>.conf**



Hình 7: Lệnh Config file

Sau khi đã chạy được câu lệnh của Vim, ta thực hiện:

- Nhấn **I** để chuyển sang chế độ insert.

- Nhập: **server-name=<tên\_file>**

Sau khi nhập xong, ta nhấn “Esc” để thoát khỏi chế độ Insert.

Để thoát và lưu file, ta nhập **“:wq”** và nhấn Enter.

A screenshot of a computer

Description automatically generated

Hình 8: Config file

**Bước 3: Cấp các quyền thực thi cho file vừa tạo**

Ta thực hiện cấp quyền đọc, viết và thực thi (read, write, execute) cho chủ sở hữu (Owner), nhóm (Group) và còn lại (Others) bằng lệnh:

**sudo chmod -R 777 /etc/rethinkdb/instances.d/<tên\_file> .conf**

A screenshot of a computer program

Description automatically generated

Hình 9: Cấp quyền cho file

**Bước 4: Chạy thử server đã tạo**

Ta tiến hành chạy thử server đã tạo bằng câu lệnh:

**rethinkdb --config-file /etc/rethinkdb/instances.d/<tên\_file>.d**

Trong câu lệnh này, để tuỳ chỉnh tên server, ta đã truyền vào câu lệnh **--config-file /etc/rethinkdb/instances.d/<tên\_file>.d** để tuỳ chỉnh tên client.

A computer screen shot of a computer program

Description automatically generated

Hình 10: Chạy server

Sau khi chạy xong câu lệnh, ta vào firefox broswer và nhập: **localhost:8080** để xem kết quả tạo.

Để dừng kết nối đến server, ta dùng lệnh **Ctrl + C** để ngắt kết nối đến server.

A screenshot of a computer

Description automatically generated

Hình 11: Kiểm tra kết quả tạo server

**Bước 5: Cài đặt python**

Ta tiến hành cài đặt môi trường để chạy các lệnh cho Rethinkdb.

Cú pháp câu lệnh cài đặt: **sudo apt install python3.10-venv.**

Trong câu lệnh, chúng ta sẽ tiến hành cài đặt phiên bản mới nhất của Python(Python3.10) để tạo môi trường.

A screenshot of a computer program

Description automatically generated

Hình 12: Cài đặt python và kết quả

Tiếp theo, ta sẽ tiến hành tạo và quản lý môi trường ảo Python. Trong hình bên dưới, tương ứng với từng dòng lệnh, ta có:

- Lệnh: **python3 -m venv ./venv.** Để tạo một môi trường ảo mới.

- Lệnh: **source venv/bin/activate**. Để kích hoạt môi trường vừa được tạo.

- Cuối cùng là lệnh: **pip install rethinkdb**. Để cài đặt gói Rethinkdb trong môi trường ảo.

A computer screen with text

Description automatically generated

Hình 13: Tạo và sử dụng môi trường python

**Bước 6: Chạy lệnh rethinkdb trên môi trường python**

Chạy server Rethinkdb như hướng dẫn ở bước 4.

Tiếp theo, ta tạo một terminal mới. Truy cập vào môi trường venv và truy cập vào python3. Sau đó ta đã có thể thực thi lệnh của Rethinkdb trên môi trường ảo python.

Trong hình bên dưới, ta có một số câu lệnh Rethinkdb đơn giản như sau:

**- from rethinkdb import r**: Đây là câu lệnh import rethinkdb để tương tác với db trên python.

**- r.connect(‘localhost’, 2810150).repl():** Câu lệnh này cho phép người dùng kết nối đến server Rethinkdb với các tham số **server = loclahost, port = 28015** và **.repl()** là câu lệnh yêu cầu phản hồi từ server.

**- r.db\_list().run():** Câu lệnh này cho phép người dùng liệt kê các database hiện đang có trên rethinkdb server.

A computer screen shot of text

Description automatically generated

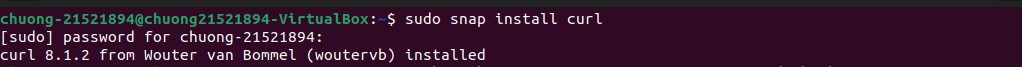
Hình 14: Chạy rethinkdb trên python

## 2.2 Cài đặt Rethinkdb cho cụm máy phân tán

Để cài đặt được cụm máy phân tán Rethinkdb, chúng ta cần để các máy chủ kết nối với nhau thông qua một hệ thống mạng hoặc là thông qua một hệ thống dịch vụ giúp kết nối. Trong phần thực hành nhóm chúng em đã sử dụng cách tiếp cận, đó là kết nối các thiết bị với nhau thông qua một dịch vụ kết nối. Ta sẽ tiến hành làm trên hai thiết bị với các bước tương tự. Các bước thực hiện lần lượt như sau:

**Bước 1: Cài đặt công cụ truyền, nhận dữ liệu.**

Ta tiến hành cài đặt curl như một công cụ để truyền và nhận dữ liệu từ các máy chủ thông qua mạng hay còn để thực hiện việc tải một số gói công cụ. Chúng ta cần tải curl như một công cụ để thực hiện các bước tiếp theo. Cú pháp câu lệnh cài đặt:  
**sudo snap install curl**



Hình 15: Tải curl

**Bước 2: Cài đặt dịch vụ kết nối mạng ảo**

Tiếp theo, thông qua curl, chúng ta sẽ tải về Zerotier. Zerotier là một dịch vụ mạng ảo giúp kết nối các thiết bị ở xa thông qua Internet như LAN. Để cài đặt qua dòng lênh, ta thực hiện như sau:

**curl -s <https://install.zerotier.com> | sudo bash.**

Trong đó **<https://install.zerotier.com>** là trang tải về mã nguồn chính thức của Zerotier.

A screenshot of a computer program

Description automatically generated

Hình 16: Cài đặt Zerotier

**Bước 3: Kết nối các máy chủ**

Ta truy cập trang web có địa chỉ: <https://my.zerotier.com> đăng nhập, tạo mạng ảo. Sau đó nhấn chọn vào Network và tìm kiếm Acces Control chuyển thành “public” để mọi máy chủ có thể kết nối vào. Cuối cùng, ta kiếm NetworkID để kết nối vào.

A screenshot of a computer

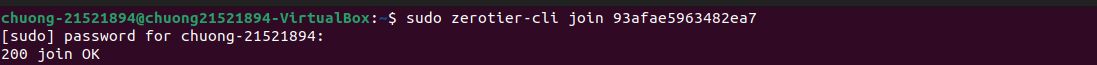
Description automatically generated

Hình 17: Giao diện Zerotier

Sau các bước thực hiện trên, ta tiến hành kết nối vào Network thông qua NetworkID, câu lệnh như sau:

**sudo zerotier-cli join <NetworkID>**

Một kết nối được coi là thành công, nếu xuất hiện thông báo: 200 join ok



Hình 18: Join Network

Sau khi đã kết nối cả hai máy trên với mạng ảo, ta tiến hành thử kết nối giữa các máy chủ thông qua câu lệnh: ping <Ip máy>.

A computer screen shot of a number

Description automatically generated

Hình 19: Kiểm tra kết nối

Kết quả ping cho thấy hai máy được kết nối tốt.

**Bước 4: Config file**

Ta tiến hành config cho hai máy thông qua câu lệnh:

**sudo nano /etc/rethinkdb/instances.d/<tên máy chủ>.conf**

- Đối với cluster, ta giữ nguyên server-name như đã thực hiện ở bước 2. đồng thời thêm:

**bind=all**



Hình 20: Lệnh Config máy cluster

A black and white screen

Description automatically generated with medium confidence

Hình 21: Config máy cluster

- Đối với máy client, làm như máy cluster nhưng thêm vào:

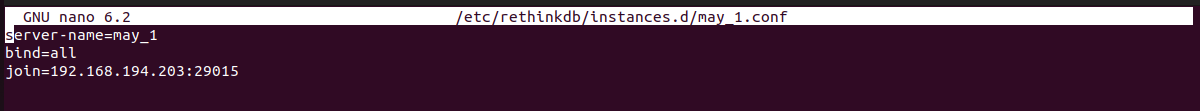
**bind=all**

**join=<ip\_cluster>:29015**

Trong phần này, ip\_cluster là ip được cung cấp bởi Zerotier.



Hình 22: Lệnh Config máy client



Hình 23: Config máy cluster

**Bước 5: Kiểm tra kết quả cuối cùng**

Sau khi đã thực hiện tất cả các bước trên, ta mở server như phần 3.1 mục 4 và truy cập loclahost để xem kết quả.

A computer screen shot of a program

Description automatically generated

Hình 24: Mở server máy cluster

A computer screen shot of a program

Description automatically generated

Hình 25: Mở server máy client

A screenshot of a computer

Description automatically generated

Hình 26: Kết quả kết nối hai máy chủ thành công.

# CHƯƠNG 3: THỰC NGHIỆM MÔ PHỎNG PHÂN TÁN

## 3.1 Mô tả bài toán đặt ra với dữ liệu

Xây dựng cơ sở dữ liệu phân tán quản lí phòng mạch tư với 2 chi nhánh nằm trong cùng khu vực tại Thành phố Hồ Chí Minh. Sau đây ta sẽ thực hiện tạo một hệ quản trị CSDL phân tán với hai chi nhánh bằng RethinkDB.

## 3.2 Mô tả cấu trúc dữ liệu sử dụng

 Cơ sở dữ liệu quản lý phòng mạch tư gồm 5 bảng:

BENHNHAN (MABENHNHAN, HOTEN, GIOITINH, NAMSINH, DIACHI)

* Quan hệ Bệnh nhân sẽ lưu trữ thông tin của bệnh nhân gồm có các thuộc tính: mã bệnh nhân, họ tên, giới tính, năm sinh, địa chỉ.

LOAITHUOC (MALOAITHUOC, TENLOAITHUOC, MACACHDUNG, SOLUONGTON, DONGIABAN, DONGIANHAP)

* Quan hệ Loại thuốc sẽ lưu trữ thông tin của loại thuốc gồm có các thuộc tính: mã loại thuốc, tên loại thuốc, mã đơn vị, mã cách dùng, số lượng tồn, đơn giá bán, đơn giá nhập.

CACHDUNG (MACACHDUNG, TENCACHDUNG)

* Quan hệ Cách dùng sẽ lưu trữ thông tin của cách dùng thuốc gồm có các thuộc tính: mã cách dùng, tên cách dùng.

HOADON (MAHOADON, MAPHIEUKHAM, TIENKHAM, TONGTIEN)

* Quan hệ Hóa đơn sẽ lưu trữ thông tin của hóa đơn khám bệnh gồm các thuộc tính: mã hóa đơn, mã phiếu khám, tiền khám, tổng tiền (gồm tiền khám bệnh và tiền thuốc nếu có).

PHIEUKHAMBENH (MAKHIEUKHAM, NGAYKHAM, MABENHNHAN, TRIEUCHUNG, TIENTHUOC)

* Quan hệ Phiếu khám bệnh sẽ lưu trữ thông tin của phiếu khám bệnh gồm các thuộc tính: mã phiếu khám, ngày khám, mã bệnh nhân, triệu chứng, tiền thuốc (tổng tiền thuốc của phiếu khám)

## 3.3 Thực nghiệm phân tán

* Kết nối phân tán giữa 2 máy **PC1** và **PC2**

A screenshot of a computer

Description automatically generated

Hình 27: Cấu hình 2 máy ảo kết nối phân tán với nhau

* **PC1** tạo CSDL “QLPMT”

**A computer screen with white text

Description automatically generated**

Hình 28: PC1 tạo DB “QLPMT”

* **PC1** tạo và thêm các bảng
  + Tạo bảng BENHNHAN  
    **A computer screen shot of white text

    Description automatically generated**

Hình 29: Tạo bảng BENHNHAN

* + Tạo các cột cho bảng BENHNHAN

A screenshot of a computer code

Description automatically generated

Hình 30: Thêm các cột cho bảng BENHNHAN

* + Thêm dữ liệu cho bảng BENHNHAN

A computer screen shot of a computer code

Description automatically generated

Hình 31: Thêm dữ liệu cho bảng BENHNHAN

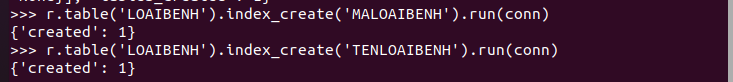
* + Tạo bảng LOAIBENH

A computer screen shot of white text

Description automatically generated

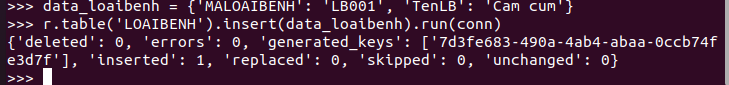
Hình 32: Tạo bảng LOAIBENH

* + Tạo các cột cho bảng LOAIBENH



Hình 33: Tạo các cột cho bảng LOAIBENH

* + Thêm dữ liệu cho bảng LOAIBENH



Hình 34: Thêm dữ liệu cho bảng LOAIBENH

* Xem Database QLPMT trên **PC1**
  + **Xem các bảng đã tạo được trên DB**

A screenshot of a computer

Description automatically generated

Hình 35: Xem các bảng đã tạo được trên DB trên PC1

* + Xem dữ liệu trong bảng
    - Bảng BENHNHAN

A screenshot of a computer

Description automatically generated

Hình 36: Bảng BENHNHAN trên PC1

* + - Bảng LOAIBENH

A screenshot of a computer

Description automatically generated

Hình 37: Bảng LOAIBENH trên PC1

* Xem Database QLPMT trên **PC2**
  + Xem các bảng đã tạo trên database

A screenshot of a computer

Description automatically generated

Hình 38: Xem các bảng đã tạo được trên DB trên PC2

* + Xem dữ liệu trong bảng
    - Bảng BENHNHAN

A screenshot of a computer

Description automatically generated

Hình 39: Bảng BENHNHAN trên PC2

* + - Bảng LOAIBENH

A screenshot of a computer

Description automatically generated

Hình 40: Bảng LOAIBENH trên PC2

* Thêm dữ liệu vào database bên **PC2**

A computer code with white text

Description automatically generated

Hình 41: Thêm dữ liệu vào database bên PC2

* Xem dữ liệu vừa thêm bên cả 2 máy, do câu lệnh được nhóm thực hiện duplicate nên xuất hiện 2 dòng ‘Viem Xoang’

A screenshot of a computer

Description automatically generated

Hình 42: Xem dữ liệu vừa thêm bên cả 2 máy

# Summary

Rethinkdb is a distributed, open-source database management system that allows storing and processing schemaless JSON documents, and pushing real-time query results to applications without polling for changes. It helps developers build real-time applications easily and quickly, with features such as flexible data storage, fine-grained access control, real-time processing, and fault tolerance.

Rethinkdb was originally created by a company of the same name and later released as an open-source project, developed by a team of database experts and over 100 contributors from around the world. Rethinkdb is built on top of JSON documents and reactive web and mobile clients, using a distributed file system, such as HDFS, to store data across multiple nodes in a cluster. Rethinkdb uses a data model called “the document-oriented database”, which allows for dynamic and flexible data access.

Rethinkdb supports multiple programming languages, such as Python, Ruby, Node.js, Java, C#, C++, Clojure, Elixir, Go, Haskell, PHP, Rust, Scala, and many others. Rethinkdb provides fine-grained access control mechanisms, allowing administrators to define access policies at the level of individual data documents. It also supports data encryption at rest, ensuring the security of stored data. Rethinkdb supports SSL/TLS to protect the connection between applications and the database. Rethinkdb supports real-time data ingest and processing, allowing for continuous updates and indexing of data, enabling near real-time analytics and decision-making. Rethinkdb allows administrators to shard and replicate data by tables or indexes, to balance load and increase fault tolerance. Rethinkdb provides a web interface to manage the cluster, monitor the status of the nodes, and perform data distribution operations.

Rethinkdb has a vibrant and supportive open-source community, with many contributors, users, and developers. Rethinkdb has many resources and documentation, such as the official website, GitHub page, API documentation, blog, and videos. Rethinkdb has many communication and exchange channels, such as Slack, Stack Overflow, Reddit, Twitter, and Google Groups.

In summary, Rethinkdb is a powerful distributed database management system designed for real-time applications with JSON documents and reactive web and mobile clients. Its origins in a company of the same name underscore its focus on innovation and scalability.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | G.Tiepolo,“https://www.packtpub.com/,”[Trực tuyến]. Available: https://www.packtpub.com/product/getting-started-with-rethinkdb/9781785887604. [Đã truy cập 22 12 2023]. |
| [2] | RethinkDB, “http://www.rethinkdb.com/,” Cloud Native Computing Foundatio, [Trực tuyến]. Available: https://dbdb.io/db/rethinkdb. [Đã truy cập 22 12 2023]. |
| [3] | RethinkDB, “https://github.com/,” [Trực tuyến]. Available: https://github.com/rethinkdb/rethinkdb?tab=readme-ov-file. [Đã truy cập 22 12 2023]. |
| [4] | Archlinux, “wiki.archlinux.org,” [Trực tuyến]. Available: https://wiki.archlinux.org/title/RethinkDB. [Đã truy cập 22 12 2023]. |
| [5] | Packtpub, “subscription.packtpub.com,” [Trực tuyến]. Available: https://subscription.packtpub.com/book/data/9781786461070/1/ch01lvl1sec13/the-rethinkdb-data-model. [Đã truy cập 22 12 2023]. |
| [6] | rethinkDB, “rethinkdb.com,” [Trực tuyến]. Available: https://rethinkdb.com/docs/introduction-to-reql/. [Đã truy cập 22 12 2023]. |
| [7] | rethinkDB, “rethinkdb.com,” [Trực tuyến]. Available: https://rethinkdb.com/docs/sql-to-reql/javascript/. [Đã truy cập 22 12 2023]. |
| [8] | rethinkdb.com, “rethinkdb.com,” [Trực tuyến]. Available: https://rethinkdb.com/docs/architecture. [Đã truy cập 22 12 2023]. |
| [9] | V. H. T. Dương, “viblo.asia,” viblo.asia, [Trực tuyến]. Available: https://viblo.asia/p/gioi-thieu-ve-dinh-ly-cap-gDVK2ONmZLj. [Đã truy cập 22 12 2023]. |