**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG THÀNH PHỐ HỒ CHÍ MINH**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**TIỂU LUẬN**

**HỌC PHẦN: HỆ QUẢN TRỊ CSDL**

**ĐỀ TÀI: Big Data and No SQL**

Sinh viên thực hiện **: Nhóm 3**

Lớp **: 09\_ĐH\_CNTT03**

Giảng viên hướng dẫn **:** **ThS. Phạm Trọng Huynh**

**TP. Hồ Chí Minh,6 tháng 12 năm 2022**

**MỞ ĐẦU**

Một thách thức công nghệ lớn mà thế giới đang phải đối mặt là quản lý và lưu trữ dữ liệu, trong đó hàng triệu dữ liệu trong thời gian gần đây đang được tạo ra với độ trễ thời gian dưới một nano giây. Do đó, việc xử lý một lượng lớn dữ liệu là một thách thức đáng kể, với sự gia tăng dân số, đòi hỏi phải có công nghệ thu thập và quản lý dữ liệu hiện đại.

(Sultana, et al, 2017) nhấn mạnh rằng nhu cầu xử lý và tạo dữ liệu nhanh chóng trong thời gian gần đây đã dẫn đến thực tế là hơn 2,6 nghìn tỷ dữ liệu đang được mã hóa hàng ngày. Họ dự đoán thêm rằng trong tương lai sẽ có sự gia tăng theo cấp số nhân về việc sử dụng và tạo ra dữ liệu trong tương lai có lưu ý đến cách mà nó đang được sử dụng ngày nay.

Nói rõ, Rõ ràng rằng việc quản lý và lưu trữ dữ liệu trong thời đại dữ liệu lớn, nơi phần mềm thông thường đang bị đẩy đến giới hạn bởi lượng dữ liệu khổng lồ, tạo ra nhu cầu thay đổi đáng kể trong việc đầu tư, thu thập và lưu trữ dữ liệu để sử dụng trong tương lai, do đó khiến các tổ chức khác nhau tham gia nỗ lực có ý thức để bảo mật và lưu giữ mọi phần dữ liệu tiềm năng.

Mặc dù dữ liệu thường không có cấu trúc, nó có thể được tạo ra từ nhiều nguồn khác nhau, chẳng hạn như đăng trên phương tiện truyền thông xã hội, nội dung đa phương tiện với các kho lưu trữ tự động. Email, truy vấn công cụ tìm kiếm, kho tài liệu quản lý nội dung, dữ liệu cảm biến thuộc nhiều loại khác nhau, trao đổi chứng khoán, hình ảnh vệ tinh, hệ thống giám sát và ứng dụng e-health.etc.

MỤC LỤC:

[Phần I : BIG DATA 1](#_Toc121877949)

[**I.** **Big Data là gì?** 1](#_Toc121877950)

[Các ứng dụng đang sử dụng Big data 1](#_Toc121877951)

[1. Ngành Ngân hàng 2](#_Toc121877952)

[2. Ngành y tế 3](#_Toc121877953)

[3.Thương mại điện tử 4](#_Toc121877954)

[4. Ngành bán lẻ 5](#_Toc121877955)

[5. Digital Marketing 6](#_Toc121877956)

[6. Ngăn chặn nội dung đen 7](#_Toc121877957)

[**3.** **Đặc điểm** 7](#_Toc121877958)

[**4.** **Quá trình hình thành** 10](#_Toc121877959)

[**5.** **Vai trò** 11](#_Toc121877960)

[**6.** **Khó khăn khi làm việc với Big Data** 12](#_Toc121877961)

[**7.** **Quy Trình Hoạt Động Của Big Data** 12](#_Toc121877962)

[**8.** **Ứng Dụng** 12](#_Toc121877963)

[1. Các công nghệ đặc biệt dành cho Big data 13](#_Toc121877964)

[1. Hệ sinh thái Hadoop 13](#_Toc121877965)

[2. Apache Spark 14](#_Toc121877966)

[3. Data lakes 14](#_Toc121877967)

[4. NoSQL Databases 14](#_Toc121877968)

[5. In-memory databases 15](#_Toc121877969)

[Phần II: No SQL 16](#_Toc121877970)

[**A.** **No SQL** 16](#_Toc121877971)

[**I.** **Cơ sở dữ liệu No SQL là gì?** 16](#_Toc121877972)

[Cơ sở dữ liệu NoSQL (phi quan hệ) hoạt động như thế nào? 16](#_Toc121877973)

[Vì sao bạn nên sử dụng cơ sở dữ liệu NoSQL? 17](#_Toc121877974)

[**II.** **Các loại cơ sở dữ liệu No SQL** 17](#_Toc121877975)

[ **Các loại cơ sở dữ liệu NoSQL** 18](#_Toc121877976)

[**III.** **Đặc điểm của No SQL** 20](#_Toc121877977)

[**IV.** **Quá Trình Hình Thành** 21](#_Toc121877978)

[**V.** **Cách thức hoạt động của CSDL No SQL** 21](#_Toc121877979)

[Vì sao chúng ta nên sử dụng NoSQL? 22](#_Toc121877980)

[Một số tính năng của NoSQL 23](#_Toc121877981)

[6. Không quan hệ (non-relational) 23](#_Toc121877982)

[7. Schema-free 23](#_Toc121877983)

[8. API đơn giản 24](#_Toc121877984)

[9. Phân tán 24](#_Toc121877985)

[**VI.** **Lợi ích của No SQL** 25](#_Toc121877986)

[Xu hướng trong tương lai? 26](#_Toc121877987)

[Ưu và nhược điểm của NoSQL 26](#_Toc121877988)

[10. Ưu điểm 26](#_Toc121877989)

[11. Nhược điểm 26](#_Toc121877990)

[**B.** **Arangodb** 27](#_Toc121877991)

[**I.** **Arangodb là gì?** 27](#_Toc121877992)

[**II.** **Đặc điểm của Arangodb** 27](#_Toc121877993)

[**III.** **Cài đặt và cấu hình cho Arangodb** 28](#_Toc121877994)

[**Phần 3: Kết Luận** 36](#_Toc121877995)

# Phần I : BIG DATA

1. **Big Data là gì?**

****

* Big Data hay “dữ liệu lớn” là thuật ngữ chỉ về các tập dữ liệu khổng lồ và phức tạp, đến mức khó có thể xử lý được bằng các phương pháp truyền thống. Doanh nghiệp sử dụng lượng dữ liệu khổng lồ này để phân tích, chuyển hóa thành thông tin quan trọng để giải quyết các vấn đề liên quan.
* Big Data thường bao gồm tập hợp dữ liệu với kích thước vượt xa khả năng của các công cụ phần mềm thông thường để thu thập, hiển thị, quản lý và xử lý dữ liệu trong một thời gian có thể chấp nhận được. Kích thước dữ liệu lớn là một mục tiêu liên tục thay đổi. Như năm 2012 thì phạm vi một vài tá terabytes tới nhiều petabytes dữ liệu. Dữ liệu lớn yêu cầu một tập các kỹ thuật và công nghệ được tích hợp theo hình thức mới để khai phá từ tập dữ liệu đa dạng, phức tạp, và có quy mô lớn.

### Các ứng dụng đang sử dụng Big data

Big data và phân tích có thể được áp dụng trong nhiều vấn đề kinh doanh và nhiều trường hợp sử dụng khác nhau. Dữ liệu lớn (Big Data) trên thực tế đang được ứng dụng vào rất nhiều lĩnh vực của nền kinh tế, tạo những chuyển biến ấn tượng, giúp tăng hiệu quả và năng suất của doanh nghiệp.

### Ngành Ngân hàng

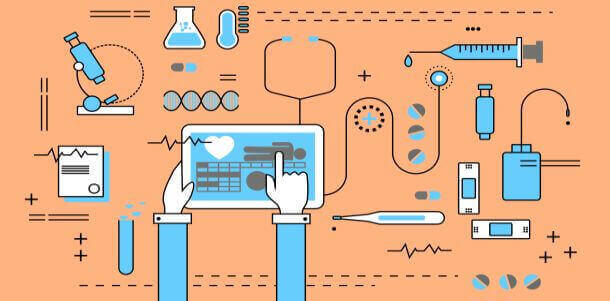


Trong hệ thống ngân hàng, Big Data đã và đang được ứng dụng hiệu quả thể hiện vai trò quan trọng của mình trong mọi hoạt động của ngân hàng: từ thu tiền mặt đến quản lý tài chính.

Ngân hàng ứng dụng Big Data như thế nào:

* Sử dụng các kỹ thuật phân cụm giúp đưa ra quyết định quan trọng. Hệ thống phân tích có thể xác định các địa điểm chi nhánh nơi tập trung nhiều nhu cầu của khách hàng tiềm năng, để đề xuất lập chi nhánh mới.
* Kết hợp nhiều quy tắc được áp dụng trong các lĩnh vực ngân hàng để dự đoán lượng tiền mặt cần thiết sẵn sàng cung ứng ở một chi nhánh tại thời điểm cụ thể hàng năm.
* Khoa học dữ liệu hiện đang là nền tảng của hệ thống ngân hàng kĩ thuật số.
* Machine learning và AI đang được nhiều ngân hàng sử dụng để phát hiện các hoạt động gian lận và báo cáo cho các chuyên viên liên quan.
* Khoa học dữ liệu hỗ trợ xử lý, lưu trữ và phân tích lượng dữ liệu khổng lồ từ các hoạt động hàng ngày và giúp đảm bảo an ninh cho ngân hàng.

### Ngành y tế



Khoa học dữ liệu đang dần khẳng định vai trò khá quan trọng trong việc cải thiện sức khỏe con người ngày nay. Big Data không chỉ được ứng dụng để xác định phương hướng điều trị mà giúp cải thiện quá trình chăm sóc sức khỏe.

Ngành y tế ứng dụng Big Data:

* Cho phép người quản lý ca dự đoán các bác sĩ cần thiết vào những thời điểm cụ thể
* Theo dõi tình trạng bệnh nhân bằng để theo dõi hồ sơ sức khỏe điện tử.
* Sử dụng các thiết bị kỹ thuật số có thể đeo, hệ thống Big Data có thể theo dõi bệnh nhân và gửi báo cáo cho các bác sĩ liên quan.
* Big Data có thể đánh giá các triệu chứng và xác định nhiều bệnh ở giai đoạn đầu.
* Có thể lưu giữ các hồ sơ nhạy cảm được bảo mật và lưu trữ lượng dữ liệu khổng lồ một cách hiệu quả.
* Các ứng dụng Big Data cũng có thể báo trước khu vực có nguy cơ bùng phát dịch như: sốt xuất huyết hoặc sốt rét.

### 3.Thương mại điện tử



Thương mại điện tử không chỉ tận hưởng những lợi ích của việc điều hành trực tuyến mà còn phải đối mặt với nhiều thách thức để đạt được các mục tiêu kinh doanh. Lý do là bởi các doanh nghiệp dù là nhỏ hay lớn, khi đã tham gia vào thị trường này đều cần đầu tư mạnh để cải tiến công nghệ. Big Data có thể tạo lợi thế cạnh tranh cho doanh nghiệp bằng cách cung cấp thông tin chuyên sâu và các bản báo cáo phân tích xu hướng tiêu dùng.

Thương mại điện tử ứng dụng Big Data:

* Có thể thu thập dữ liệu và yêu cầu của khách hàng ngay cả trước khi khách thực sự bắt đầu giao dịch.
* Tạo ra một mô hình tiếp thị hiệu suất cao.
* Nhà quản lý trang thương mại điện tử có thể xác định các sản phẩm được xem nhiều nhất và tối ưu thời gian hiển thị của các trang sản phẩm này.
* Đánh giá hành vi của khách hàng và đề xuất các sản phẩm tương tự. Điều này làm tăng khả năng bán hàng, từ đó tạo ra doanh thu cao hơn.
* Nếu bất kỳ sản phẩm nào được thêm vào giỏ hàng nhưng cuối cùng không được khách hàng mua, Big Data có thể tự động gửi code khuyến mại cho khách hàng cụ thể đó.
* Các ứng dụng Big Data còn có thể tạo một báo cáo tùy chỉnh theo các tiêu chí: độ tuổi, giới tính, địa điểm của khách truy cập, v.v.
* Xác định các yêu cầu của khách hàng, những gì họ muốn và tập trung vào việc cung cấp dịch vụ tốt nhất để thực hiện nhu cầu của họ.
* Phân tích hành vi, sự quan tâm của khách hàng và theo xu hướng của họ để tạo ra các sản phẩm hướng đến khách hàng.
* Cung cấp các sản phẩm tốt hơn với chi phí thấp hơn.
* Có thể thu thập nhiều dữ liệu về hành vi khách hàng để thiết kế mô hình tiếp thị tối ưu dành được tùy biến theo đối tượng hoặc nhóm đối tượng, tăng khả năng bán hàng.
* Tìm ra sự tương đồng giữa khách hàng và nhu cầu của họ. Từ đó, việc nhắm mục tiêu các chiến dịch quảng cáo có thể được tiến hành dễ dàng hơn dựa trên những phân tích đã có trước đó.

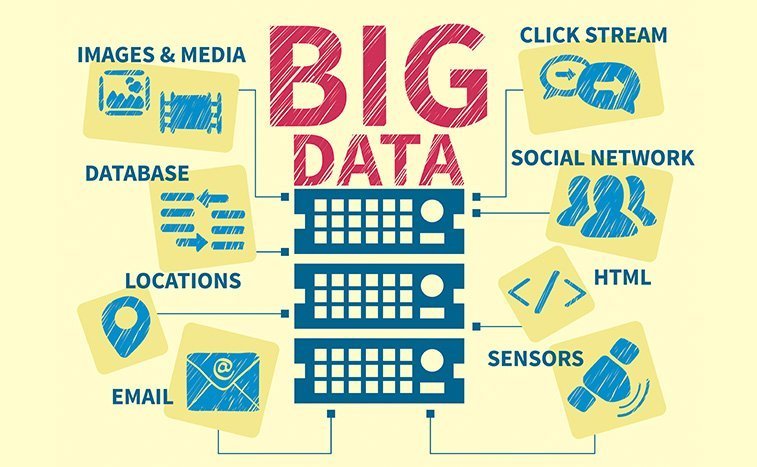
### 4. Ngành bán lẻ

Big Data mang lại cơ hội cho lĩnh vực bán lẻ bằng cách phân tích thị trường cạnh tranh và sự quan tâm của khách hàng. Nó giúp xác định hành trình trải nghiệm, xu hướng mua sắm và sự hài lòng của khách hàng bằng cách thu thập dữ liệu đa dạng. Từ những dữ liệu thu thập được có thể cải thiện hiệu suất và hiệu quả bán hàng.

Ngành bán lẻ ứng dụng Big Data:

* Big data giúp nhà quản lý xây dựng mô hình chi tiêu của từng khách hàng.
* Với sự trợ giúp của các phân tích dự đoán, ngành công nghiệp có thể so sánh tỷ lệ cung – cầu và có thể tránh tiếp tục tung ra thị trường các sản phẩm không được hầu hết khách hàng đón nhận.
* Ngành bán lẻ có thể xác định vị trí bố trí sản phẩm trên kệ hàng tùy thuộc vào thói quen mua hàng và nhu cầu của khách hàng và đưa ra các chiến lược kinh doanh mới để cải thiện.
* Kết hợp phân tích cùng lúc các dữ liệu về thời điểm, dữ liệu giao dịch, dữ liệu truyền thông xã hội, dự báo thời tiết để xác định chính xác nhất sản phẩm phù hợp để luôn sẵn sàng cung ứng cho khách hàng.

### 5. Digital Marketing



Digital Marketing là chìa khóa để cánh cửa thành công cho bất kỳ doanh nghiệp nào. Giờ đây, không chỉ các công ty lớn có thể điều hành các hoạt động quảng cáo tiếp thị mà cả các doanh nhân nhỏ cũng có thể chạy các chiến dịch quảng cáo thành công trên các nền tảng truyền thông xã hội và quảng bá sản phẩm của họ. Big Data đã tiếp sức cho Digital Marketing phát triển thực sự mạnh mẽ, và nó đã trở thành một phần không thể thiếu của bất kỳ doanh nghiệp nào.

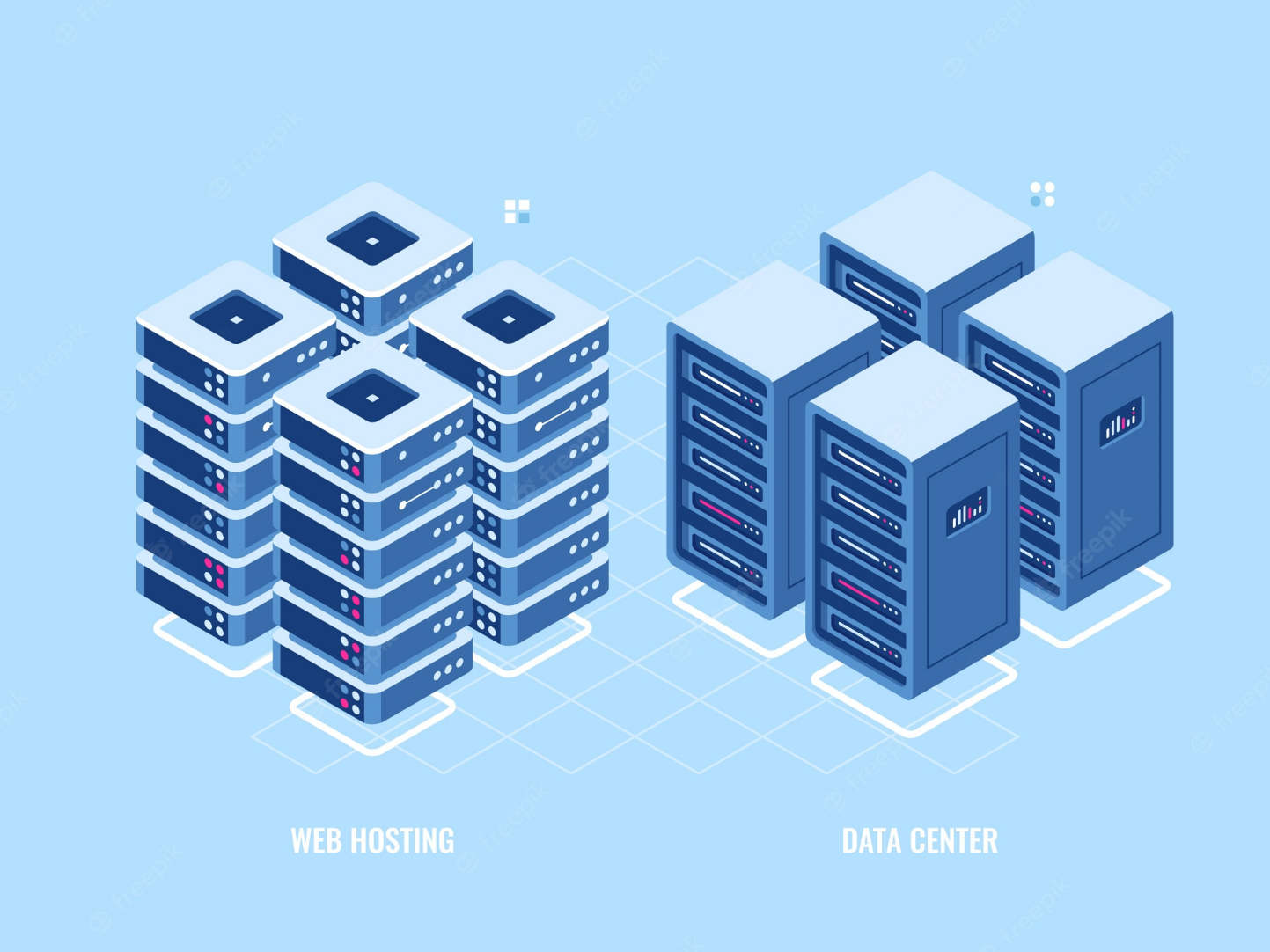
Digital Marketing ứng dụng Big Data:

* Phân tích thị trường, đối thủ cạnh tranh và đánh giá mục tiêu kinh doanh. Điều này giúp cho doanh nghiệp xác định rõ hơn, đâu là cơ hội tốt để tiếp tục tiến hành các kế hoạch kinh doanh tiếp theo.
* Có thể xác định người dùng trên các phương tiện truyền thông xã hội và nhắm mục tiêu cho họ dựa trên nhân khẩu học, giới tính, thu nhập, tuổi tác và sở thích.
* Tạo báo cáo sau mỗi chiến dịch quảng cáo bao gồm hiệu suất, sự tham gia của khán giả và những gì có thể được thực hiện để tạo kết quả tốt hơn.
* Khoa học dữ liệu được sử dụng cho các khách hàng nhắm mục tiêu và nuôi dưỡng chu trình khách hàng.
* Tập trung vào các chủ đề được tìm kiếm cao và tư vấn cho các chủ doanh nghiệp thực hiện chúng trên chiến lược nội dung để xếp hạng trang web doanh nghiệp trên cao hơn trên google (SEO).
* Có thể tạo đối tượng tương tự bằng cách sử dụng cơ sở dữ liệu đối tượng hiện có để nhắm mục tiêu các khách hàng tương tự và kiếm được lợi nhuận.

### **6. Ngăn chặn nội dung đen**

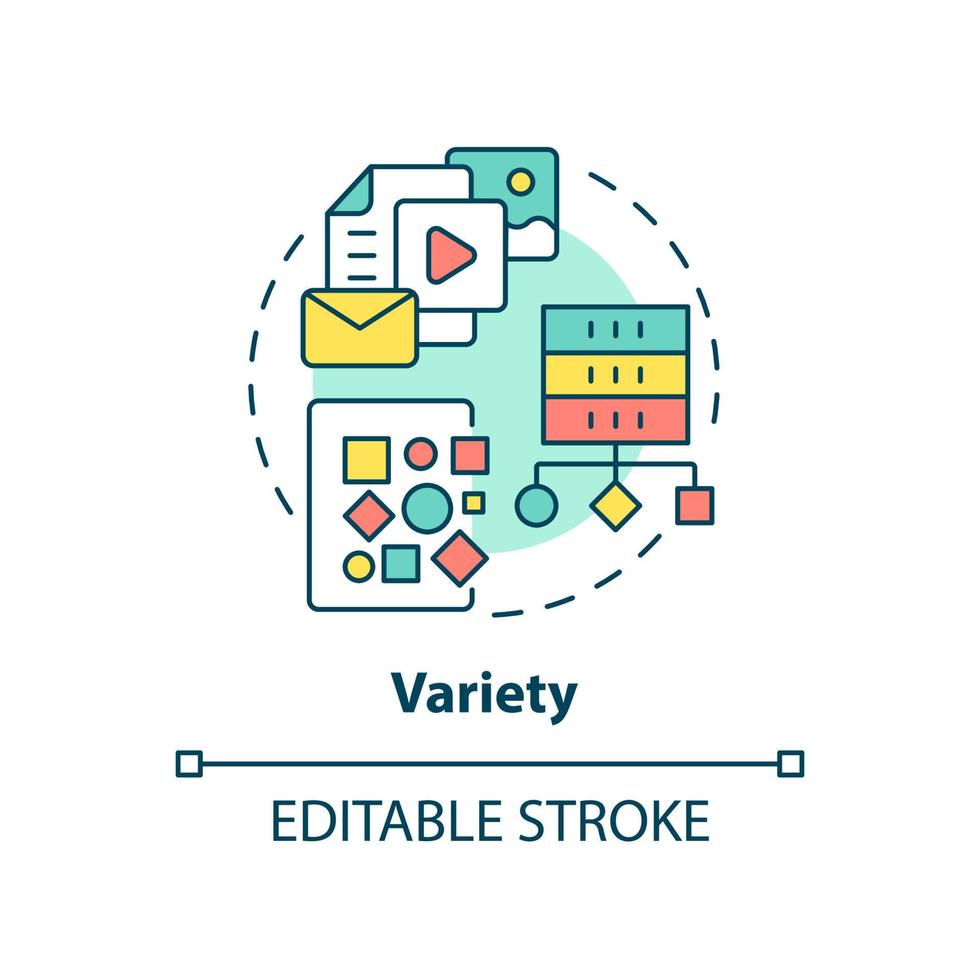
Ví dụ cụ thể như là Extension (Chrome, Firefox, Safari…) Có nhiều addon phục vụ cho việc content filtering miễn phí sử dụng Bigdata để thu thập và dự đoán xem nội dung đó có phù hợp không. Ví dụ chức năng Ad Block nhanh chóng block các banner, pop ups, video ads gây phiền nhiễu một lần và mãi mãi. Sau đó nó lập tức thu thập và gửi về server blacklist những yếu tố này. Data càng nhiều thì tỷ lệ nhận diện và block ngày càng chính xác.

1. **Đặc điểm**

* **Volume (khối lượng dữ liệu):** Số lượng dữ liệu được tạo ra và lưu trữ. Kích thước của dữ liệu xác định giá trị và tiềm năng insight- và liệu nó có thể thực sự được coi là dữ liệu lớn hay không.
* 
* **Velocity (vận tốc xử lý):** Trong trường hợp này nghĩa là tốc độ các dữ liệu được tạo ra và xử lý để đáp ứng các nhu cầu và thách thức trên con đường tăng trưởng và phát triển.



* **Variety (dữ liệu đa dạng):**Các dạng và kiểu của dữ liệu. Dữ liệu được thu thập từ nhiều nguồn khác nhau và các kiểu dữ liệu cũng có rất nhiều cấu trúc khác nhau.



* **Veracity (Tính xác thực)**

Chất lượng của dữ liệu thu được có thể khác nhau rất nhiều, ảnh hưởng đến sự phân tích chính xác.

Nhà máy và các hệ thống không thực-ảo có thể có một hệ thống 6C bao gồm:

• Kết nối (cảm biến và mạng)

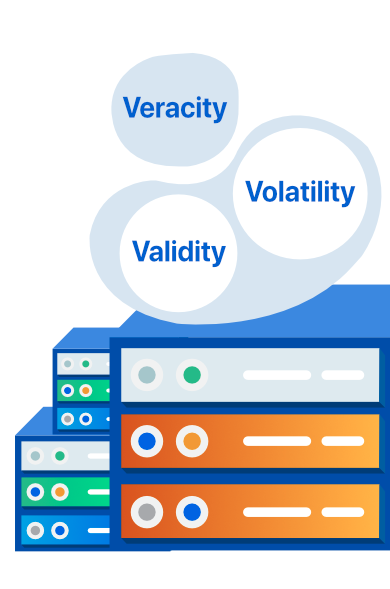
• Đám mây (tính toán và dữ liệu theo yêu cầu)

• Nội dung ảo (mẫu và bộ nhớ)

• Nội dung / ngữ cảnh (ý nghĩa và tương quan)

• Cộng đồng (chia sẻ và cộng tác)

• Tuỳ chỉnh (cá nhân hoá và giá trị)



1. **Quá trình hình thành**

Big Data thực chất đã hình thành từ khoảng thập kỷ 80 - 90 của thế kỷ XX. Năm 1984, tập đoàn Teradata đưa ra thị trường hệ thống xử lý dữ liệu song song DBC 1012. Các hệ thống xử lý của Teradata là một trong những hệ thống đầu tiên lưu trữ và phân tích đến 1 terabyte dữ liệu vào năm 1992.



1. **Vai trò**
2. Hiểu và nhắm đúng khách hàng mục tiêu
3. Định lượng và tối ưu hóa hiệu suất cá nhân
4. Phòng chống an ninh, giảm thiểu rủi ro
5. Tối ưu hóa giá cả
6. Nắm bắt các giao dịch tài chính



1. **Khó khăn khi làm việc với Big Data**

* Thách thức lớn nhất khi sử dụng Big Data đó là khối lượng quá lớn và tăng nhanh
* chi phí đầu tư cho nó cũng không hề nhỏ
* không có nhiều người biết tận dụng nó
* việc sử dụng Big Data cũng bị chỉ trích vì có trường hợp xâm phạm đến quyền riêng tư của người dùng. Tại một số quốc gia thì đây là một việc vi phạm pháp luật.
* quá phụ thuộc vào dữ liệu từ Big Data có thể gây ra sự chủ quan, trong một số trường hợp có thể ảnh hưởng đến kết quả cuối cùng.

1. **Quy Trình Hoạt Động Của Big Data**
2. Xây dựng chiến lược Big Data
3. Xác định các nguồn Big Data :

- Dữ liệu truyền trực tiếp

- Social Media (Facebook, YouTube, Instagram,...)

-  Dữ liệu có sẵn được công bố

- Nguồn dữ liệu còn lại ( từ khách hàng, nhà cung cấp, dữ liệu đám mây)

1. Truy cập, quản lý và lưu trữ Big Data
2. Tiến hành phân tích dữ liệu
3. Dựa trên dữ liệu đưa ra quyết định
4. **Ứng Dụng**
5. Tài chính
6. Chăm sóc sức khỏe
7. Sản xuất
8. Giáo dục
9. Truyển thông
10. Công nghệ

## Các công nghệ đặc biệt dành cho Big data

Ngoài cơ sở hạ tầng IT được sử dụng cho dữ liệu nói chung, có một số công nghệ cụ thể dành cho big data mà cơ sở hạ tầng IT của bạn nên hỗ trợ.

### Hệ sinh thái Hadoop

[Hadoop](https://topdev.vn/blog/hadoop-la-gi/) là một trong những công nghệ liên quan chặt chẽ nhất với big data. Dự án Apache Hadoop phát triển phần mềm mã nguồn mở cho máy tính có khả năng mở rộng và phân tán.

Thư viện phần mềm Hadoop là một khuôn mẫu cho phép xử lý phân tán các bộ dữ liệu lớn trên các nhóm máy tính sử dụng các mô hình lập trình đơn giản. Nó được thiết kế để mở rộng từ một máy chủ duy nhất sang hàng ngàn máy khác, mỗi máy cung cấp tính toán và lưu trữ cục bộ.

**Dự án bao gồm rất nhiều phần:**

* Hadoop Common, các tiện ích phổ biến hỗ trợ các phần Hadoop khác
* Hadoop Distributed File System, cung cấp khả năng truy cập dữ liệu ứng dụng cao
* Hadoop YARN, một khuôn mẫu cho kế hoạch làm việc và quản lý tài nguyên cụm
* Hadoop MapReduce, một hệ thống dựa trên YARN để xử lý song song bộ dữ liệu lớn.

### Apache Spark

Một phần của hệ sinh thái Hadoop, Apache Spark là một khuôn mẫu tính toán cụm nguồn mở được sử dụng làm công cụ xử lý big data trong Hadoop.

Spark đã trở thành một trong những khuôn mẫu xử lý big data quan trọng, và có thể được triển khai theo nhiều cách khác nhau. Nó cung cấp các phương thức hỗ trợ đối với [Java](https://topdev.vn/blog/lap-trinh-java/), [Scala](https://topdev.vn/blog/scala-la-gi/), [Python](https://topdev.vn/blog/su-dung-python-de-lam-gi/) (đặc biệt là Anaconda Python distro ), và ngôn ngữ lập trình R ( R đặc biệt phù hợp với big data ) và hỗ trợ [SQL](https://topdev.vn/blog/sql-la-gi/), streaming data, [machine learning](https://topdev.vn/blog/tan-man-ve-machine-learning/) và xử lý đồ thị.

### Data lakes

Data lakes là các kho lưu trữ chứa khối lượng dữ liệu thô rất lớn ở định dạng gốc của nó cho đến khi những người dùng doanh nghiệp cần dữ liệu.

Các yếu tố giúp tăng trưởng data lakes là những phong trào kỹ thuật số và sự phát triển của IoT. Các data lakes được thiết kế để giúp người dùng dễ dàng truy cập vào một lượng lớn dữ liệu khi có nhu cầu.

### NoSQL Databases

Các cơ sở dữ liệu SQL thông thường được thiết kế cho các transaction đáng tin cậy và các truy vấn ngẫu nhiên.

Nhưng chúng có những hạn chế như giản đồ cứng nhắc làm cho chúng không phù hợp với một số loại ứng dụng. Cơ sở dữ liệu NoSQL nêu ra những hạn chế, và lưu trữ và quản lý dữ liệu theo những cách cho phép tốc độ hoạt động cao và sự linh hoạt tuyệt vời.

Nhiều cơ sở dữ liệu đã được phát triển bởi các công ty để tìm cách tốt hơn để lưu trữ nội dung hoặc xử lý dữ liệu cho các trang web lớn. Không giống như các cơ sở dữ liệu SQL, nhiều cơ sở dữ liệu NoSQL có thể được mở rộng theo chiều ngang trên hàng trăm hoặc hàng ngàn máy chủ.

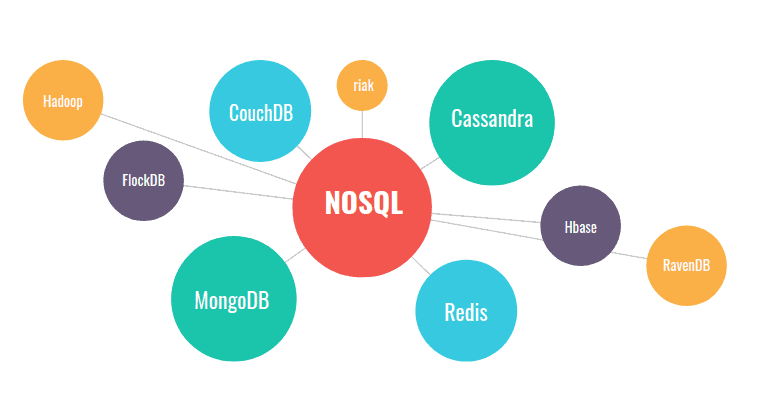
### In-memory databases

Cơ sở dữ liệu trong bộ nhớ (IMDB) là một hệ thống quản lý cơ sở dữ liệu chủ yếu dựa vào bộ nhớ chính (Ram), thay vì HDD, để lưu trữ dữ liệu. Cơ sở dữ liệu trong bộ nhớ nhanh hơn các cơ sở dữ liệu được tối ưu hóa trong đĩa, một điểm quan trọng để sử dụng phân tích big data và tạo ra các kho dữ liệu và các siêu dữ liệu.

# Phần II: No SQL

1. **No SQL**
2. **Cơ sở dữ liệu No SQL là gì?**

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. Trang này có các tài nguyên giúp bạn hiểu thêm về cơ sở dữ liệu NoSQL và bắt đầu sử dụng.



# Cơ sở dữ liệu NoSQL (phi quan hệ) hoạt động như thế nào?

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.

Hãy xem ví dụ về mô hình sơ đồ dành cho cơ sở dữ liệu sách đơn giản:

* Trong cơ sở dữ liệu quan hệ, hồ sơ về một cuốn sách thường được phân tách (hay còn gọi là "chuẩn hóa") và lưu trữ trong các bảng tách biệt nhau, còn mối quan hệ được quy định bằng các ràng buộc khóa ngoại và khóa chính. Trong ví dụ này, bảng Sách có các cột cho ISBN, Tên sách và Số phiên bản, bảng Tác giả có các cột cho ID tác giả và Tên tác giả và cuối cùng, bảng Tác giả–ISBN có các cột cho ID tác giả và ISBN. Mô hình quan hệ được thiết kế để cho phép các cơ sở dữ liệu này thực thi tính toàn vẹn tham chiếu giữa nhiều bảng trong cơ sở dữ liệu, được chuẩn hóa để giảm dư thừa và thường được tối ưu hóa cho mục đích lưu trữ.
* Trong cơ sở dữ liệu NoSQL, hồ sơ về một cuốn sách thường được lưu trữ dưới dạng văn bản [JSON](http://json.org/). Với từng quyển sách, mục, ISBN, Tên sách, Số phiên bản, Tên tác giả và ID tác giả được lưu trữ dưới dạng thuộc tính trong một văn bản duy nhất. Trong mô hình này, dữ liệu được tối ưu hóa cho việc phát triển trực quan và khả năng thay đổi quy mô theo chiều ngang.

Vì sao bạn nên sử dụng cơ sở dữ liệu NoSQL?

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.

1. **Các loại cơ sở dữ liệu No SQL**

Quá trình phát triển đã cho ra đời bốn loại NoSQL chính, cụ thể:

* **Document databases**: Đây là loại cơ sở dữ liệu tài liệu lưu trữ và truy xuất dữ liệu dưới dạng một cặp khóa giá trị. Các giá trị đa dạng chẳng hạn như số, chuỗi, đối tượng…
* **Key – Value databases**: NoSQL loại này khá đơn giản vì mỗi mục của nó chỉ chứa key và value.
* **Wide-column stores**: Dữ liệu ở NoSQL loại này được lưu trữ trong bảng, cột động và hàng.
* **Graph databases**: Dữ liệu ở Graph databases được lưu trữ trong các nút và các cạnh.
* **Các loại cơ sở dữ liệu NoSQL**
* 
* **Khóa–giá trị:** Cơ sở dữ liệu khóa–giá trị có khả năng phân mảnh cao và cho phép thay đổi quy mô theo chiều ngang ở các quy mô lớn mà các loại hình cơ sở dữ liệu khác không thể làm được. Mô hình dữ liệu khóa–giá trị được dùng cho trò chơi, công nghệ quảng cáo và đặc biệt thích hợp cho IoT. [Amazon DynamoDB](https://aws.amazon.com/dynamodb/) được thiết kế để có độ trễ ổn định chỉ vài mili giây cho khối lượng công việc thuộc quy mô bất kỳ. Hiệu năng ổn định này là lý do chính để di chuyển [tính năng Stories của Snapchat](https://www.youtube.com/watch?v=WUleQzu9l_8) (kể cả khối lượng công việc ghi lưu trữ lớn nhất của Snapchat) sang DynamoDB.
* 
* **Tài liệu:** Trong mã ứng dụng, dữ liệu thường được biểu diễn dưới dạng một đối tượng hoặc văn bản dạng JSON vì đây là mô hình dữ liệu hiệu quả và trực quan cho các nhà phát triển. Cơ sở dữ liệu văn bản giúp nhà phát triển dễ dàng lưu trữ và truy vấn dữ liệu trong một cơ sở dữ liệu bằng cách sử dụng cùng một định dạng mô hình văn bản họ sử dụng trong mã ứng dụng của mình. Tính chất linh hoạt, bán cấu trúc và phân cấp của các văn bản và cơ sở dữ liệu văn bản cho phép chúng phát triển phù hợp với yêu cầu của ứng dụng. Mô hình văn bản phát huy hiệu quả với danh mục, hồ sơ người dùng và hệ thống quản lý nội dung, nơi từng văn bản là duy nhất và phát triển theo thời gian. [Amazon DocumentDB (với khả năng tương thích MongoDB)](https://aws.amazon.com/documentdb/) và MongoDB là các cơ sở dữ liệu văn bản phổ biến cung cấp các API mạnh mẽ và trực quan để phát triển linh hoạt và lặp lại.
* 
* **Đồ thị:** Mục đích của cơ sở dữ liệu đồ thị là giúp việc dựng và chạy ứng dụng hoạt động với các bộ dữ liệu có khả năng kết nối cao trở nên dễ dàng. Cơ sở dữ liệu đồ thị thường được sử dụng cho các đồ thị tri thức, mạng xã hội, công cụ đề xuất và phát hiện lừa đảo. [Amazon Neptune](https://aws.amazon.com/neptune/) là dịch vụ cơ sở dữ liệu đồ thị được quản lý đầy đủ. Neptune hỗ trợ cả mô hình Đồ thị thuộc tính lẫn Framework mô tả tài nguyên (RDF), cung cấp cho người dùng lựa chọn hai API đồ thị: TinkerPop và RDF/SPARQL. Các cơ sở dữ liệu đồ thị phổ biến gồm có Neo4j và Giraph.
* 
* **Trong bộ nhớ:** Ứng dụng trò chơi và công nghệ quảng cáo đã sử dụng loại Cơ sở dữ liệu này trong nhiều trường hợp như bảng xếp hạng, kho lưu trữ phiên và phân tích thời gian thực yêu cầu thời gian phản hồi đạt mức micro giây và lưu lượng có thể đạt mức đỉnh điểm vào bất kỳ thời điểm nào. [Amazon MemoryDB for Redis](http://aws.amazon.com/memorydb) là một dịch vụ cơ sở dữ liệu trong bộ nhớ, bền, tương thích với Redis có chức năng đảm bảo độ trễ của thao tác đọc tính bằng micro giây, độ trễ của thao tác ghi chưa đến mười mili giây và độ bền trên nhiều vùng sẵn sàng. MemoryDB được xây dựng với mục đích đem lại độ bền và hiệu suất cực nhanh để bạn có thể sử dụng làm cơ sở dữ liệu chính cho các ứng dụng vi dịch vụ, hiện đại. [Amazon ElastiCache](http://aws.amazon.com/elasticache) là dịch vụ bộ nhớ đệm trong bộ nhớ được quản lý hoàn toàn, tương thích với cả Redis và Memcached để phục vụ khối lượng công việc có độ trễ thấp, thông lượng cao. Các khách hàng yêu cầu khả năng phản hồi theo thời gian thực cho ứng dụng của mình như [Tinder](https://aws.amazon.com/blogs/database/building-resiliency-at-scale-at-tinder-with-amazon-elasticache/) phụ thuộc vào những kho lưu trữ dữ liệu trong bộ nhớ hơn là lưu trữ dữ liệu trên đĩa. [Amazon DynamoDB Accelerator](https://aws.amazon.com/dynamodb/dax/) (DAX) là ví dụ khác về kho dữ liệu được xây dựng chuyên dụng. DAX giúp DynamoDB đọc nhanh hơn rất nhiều.
* 
* **Tìm kiếm:** Nhiều ứng dụng xuất nhật ký để giúp nhà phát triển khắc phục sự cố. [Dịch vụ OpenSearch của Amazon](https://aws.amazon.com/opensearch-service/) được xây dựng nhằm mục đích cung cấp hình ảnh trực quan và phân tích gần như trong thời gian thực về các dữ liệu do máy tạo bằng cách lập chỉ mục, tập hợp và tìm kiếm trong các nhật ký và chỉ số bán cấu trúc. Dịch vụ OpenSearch của Amazon cũng là công cụ tìm kiếm mạnh mẽ, có hiệu suất cao cho các trường hợp sử dụng tìm kiếm toàn bộ văn bản. [Expedia](https://www.youtube.com/watch?v=oJUpUQ_yNVw) đang sử dụng trên 150 miền Dịch vụ OpenSearch của Amazon, 30 TB dữ liệu và 30 tỷ văn bản cho nhiều chức năng đóng vai trò quan trọng, từ theo dõi và khắc phục sự cố trong khâu vận hành đến tối ưu hóa quá trình theo dấu và định giá chồng ứng dụng được phân phối.
* NoSQL đang dần nổi lên như một thế lực trong giới lập trình. Nhà nhà quảng cáo NoSQL, người người sử dụng NoSQL. MEAN stack (MongoDB, Express, AngularJS, NodeJS) đang dần lấn lướt, thay thế cho LAMP stack (Linux, Apache, MySQL, PHP/Python) đã lỗi thời.

1. **Đặc điểm của No SQL**

* NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin.
* Mô hình phân tán dưới sự kiểm soát phần mềm.
* Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau.
* Một truy vấn sẽ được gửi tới nhiều máy cùng lúc, do đó khi một máy nào đó không phục vụ được sẽ không ảnh hưởng lắm đến chất lượng trả về kết quả.
* Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu.
* Tính nhất quán không theo thời gian thực: Sau mỗi thay đổi CSDL, không cần tác động ngay đến tất cả các CSDL liên quan mà được lan truyền theo thời gian.

1. **Quá Trình Hình Thành**

Lịch sử Thuật ngữ NoSQL được sử dụng bởi Carlo Strozzi vào năm 1998 để đặt tên cho cơ sở dữ liệu quan hệ mã nguồn mở Strozzi NoSQL nhỏ gọn của mình, mà không tiết lộ giao diện SQL tiêu chuẩn, nhưng là vẫn còn là kiểu quan hệ.

1. **Cách thức hoạt động của CSDL No SQL**

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.

Hãy xem ví dụ về mô hình sơ đồ dành cho cơ sở dữ liệu sách đơn giản:

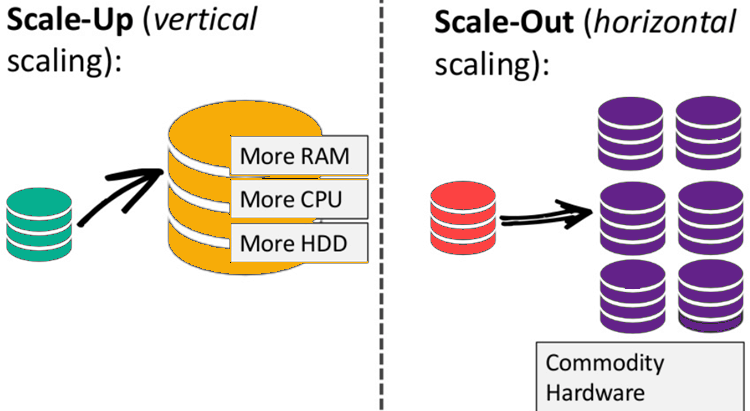
* Trong cơ sở dữ liệu quan hệ, hồ sơ về một cuốn sách thường được phân tách (hay còn gọi là “chuẩn hóa”) và lưu trữ trong các bảng tách biệt nhau, còn mối quan hệ được quy định bằng các ràng buộc khóa ngoại và khóa chính. Trong ví dụ này, bảng **Sách**có các cột cho **ISBN**, **Tên sách** và **Số phiên bản**, bảng **Tác giả**có các cột cho **ID tác giả**và **Tên tác giả** và cuối cùng, bảng **Tác giả–ISBN** có các cột cho **ID tác giả**và **ISBN**. Mô hình quan hệ được thiết kế để cho phép các cơ sở dữ liệu này thực thi tính toàn vẹn tham chiếu giữa nhiều bảng trong cơ sở dữ liệu, được chuẩn hóa để giảm dư thừa và thường được tối ưu hóa cho mục đích lưu trữ.
* Trong cơ sở dữ liệu NoSQL, hồ sơ về một cuốn sách thường được lưu trữ dưới dạng văn bản [JSON](http://json.org/). Với từng quyển sách, mục, **ISBN**, **Tên sách**, **Số phiên bản**, **Tên tác giả** và **ID tác giả**được lưu trữ dưới dạng thuộc tính trong một văn bản duy nhất. Trong mô hình này, dữ liệu được tối ưu hóa cho việc phát triển trực quan và khả năng thay đổi quy mô theo chiều ngang.

## Vì sao chúng ta nên sử dụng NoSQL?

Khái niệm cơ sở dữ liệu NoSQL trở nên phổ biến với những công ty công nghệ lớn như Google, Facebook, Amazon, …, những công ty cần phải xử lý khối lượng dữ liệu khổng lồ. Thời gian phản hồi của hệ thống trở nên chậm nếu như bạn sử dụng RDBMS truyền thống để truy vấn khối lượng lớn dữ liệu.

Để giải quyết vấn đề này, chúng ta có thể “mở rộng quy mô” hệ thống của mình bằng cách nâng cấp phần cứng hiện có của chúng ta. Quá trình này là tốn kém.

Giải pháp thay thế cho vấn đề này là phân phối tải cơ sở dữ liệu trên nhiều máy chủ bất cứ khi nào tải tăng lên. Phương pháp này được gọi là “mở rộng quy mô”.



Qua đó chúng ta có thể thấy 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.

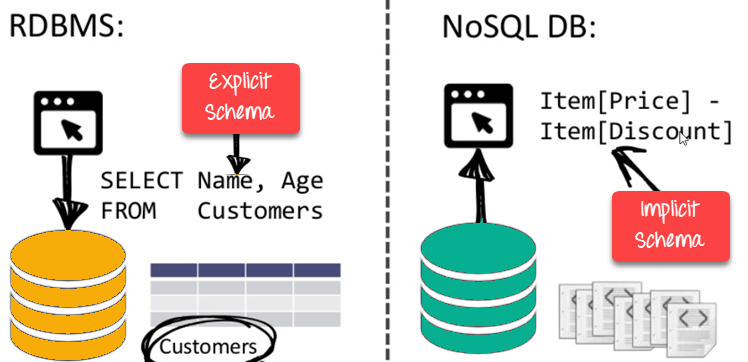
## Một số tính năng của NoSQL

### Không quan hệ (non-relational)

* Cơ sở dữ liệu NoSQL không tuân theo mô hình quan hệ
* Không yêu cầu ánh xạ quan hệ đối tượng và chuẩn hóa dữ liệu
* Không có các tính năng phức tạp như query language, query planner, toàn vẹn tham chiếu, ACID

### Schema-free

* Cơ sở dữ liệu NoSQL có thể không có schema hoặc có một schema linh hoạt
* Không yêu cầu bất kỳ loại định nghĩa nào về schema dữ liệu
* Cung cấp cấu trúc dữ liệu không đồng nhất trong cùng một miền

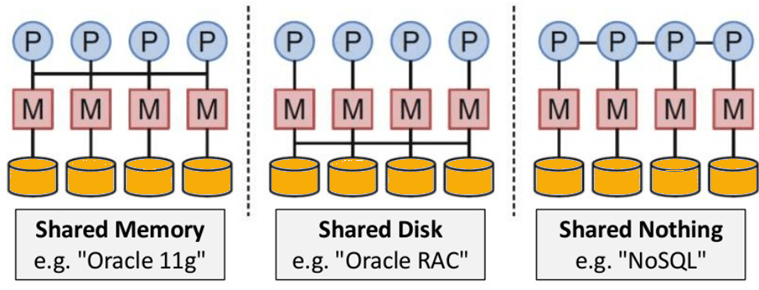


### API đơn giản

* Cung cấp giao diện dễ sử dụng để lưu trữ và truy vấn dữ liệu được cung cấp
* API cho phép các phương pháp lựa chọn và thao tác dữ liệu cấp thấp
* Các giao thức dựa trên văn bản chủ yếu được sử dụng với HTTP REST với JSON
* Hầu như không sử dụng các tiêu chuẩn cho ngôn ngữ truy vấn NoSQL

### Phân tán

* Nhiều cơ sở dữ liệu NoSQL có thể được thực thi theo kiểu phân tán
* Cung cấp khả năng tự động mở rộng quy mô và khả năng vượt qua lỗi
* Chỉ cung cấp tính nhất quán cuối cùng
* Shared Nothing Architecture. Điều này cho phép phối hợp ít hơn và phân tán cao hơn.



1. **Lợi ích của No SQL**

Cơ sở dữ liệu NoQuery có khả năng mở rộng và mô hình dữ liệu của NoSQL giải quyết một số vấn đề mà mô hình quan hệ không giải quyết được. Cụ thể một số lợi ích của NoSQL là:

* **Lược đồ linh hoạt**: NoSQL có thể làm với dữ liệu dạng không có cấu trúc. Nhờ đó, sở dữ liệu NoSQL là lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.
* **Có khả năng co dãn (Elasticity):** Việc thay đổi cấu trúc dữ liệu (Thêm/xóa trường hoặc bảng) rất dễ dàng.
* **Hiệu năng cao**: Cơ sở dữ liệu NoSQL được tối ưu hóa theo mô hình dữ liệu (ví dụ như văn bản, khóa–giá trị và đồ thị) giúp truy vấn dễ dàng hơn so với cơ sở dữ liệu quan hệ RDBMS.
* **Có khả năng mở rộng**: Do không đặt nặng tính ACID của transactions và tính nhất quán của dữ liệu nên NoSQL DB có thể mở rộng, chạy trên nhiều máy một cách dễ dàng.

### Xu hướng trong tương lai?

NoSQL bỏ qua tính toàn vẹn của dữ liệu và ACID transaction của RDBMS để đổi lấy hiệu suất nhanh và khả năng mở rộng (scalability). Do đó, NoSQL được ứng dụng nhiều trong các dự án Big Data, các dự án Real-time,... với lượng dữ liệu khổng lồ hoặc không có cấu trúc cụ thể. Đồng thời do vậy nên NoSQL sẽ không thể thay thế hoàn toàn RDBMS. Tuy nhiên, thay vì "one size fits all" như trước kia thì nay các dự án sẽ sử dụng không chỉ RDBMS hoặc NoSQL mà có thể là cả hai. Chính vì lẽ dó, việc trau dồi kiến thức SQL cũng như tìm hiểu thêm về NoSQL là vô cùng cần thiết với mọi lập trình viên.

## Ưu và nhược điểm của NoSQL

### Ưu điểm

* Khả năng truy vấn dữ liệu lớn
* Mở rộng dễ dàng
* Nó cung cấp hiệu suất nhanh chóng và khả năng mở rộng theo chiều ngang.
* Có thể xử lý dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc với hiệu quả như nhau
* Lập trình hướng đối tượng dễ sử dụng và linh hoạt
* Cơ sở dữ liệu NoSQL không cần máy chủ hiệu suất cao chuyên dụng
* Hỗ trợ nền tảng và ngôn ngữ chính dành cho nhà phát triển

### Nhược điểm

* Không có quy tắc tiêu chuẩn hóa
* Khả năng truy vấn hạn chế
* Cơ sở dữ liệu và công cụ RDBMS tương đối hoàn thiện
* Nó không cung cấp bất kỳ khả năng cơ sở dữ liệu truyền thống nào, như tính nhất quán khi nhiều transactions được thực hiện đồng thời.
* Khi khối lượng dữ liệu tăng lên, rất khó để duy trì các giá trị duy nhất vì các khóa trở nên khó khăn
* Các tùy chọn mã nguồn mở nên không quá phổ biến đối với các doanh nghiệp.

1. **Arangodb**
2. **Arangodb là gì?**

là**một hệ thống cơ sở dữ liệu NoSQL mã nguồn mở được quản lý dễ dàng thông qua giao diện web tích hợp hoặc giao diện dòng lệnh.**

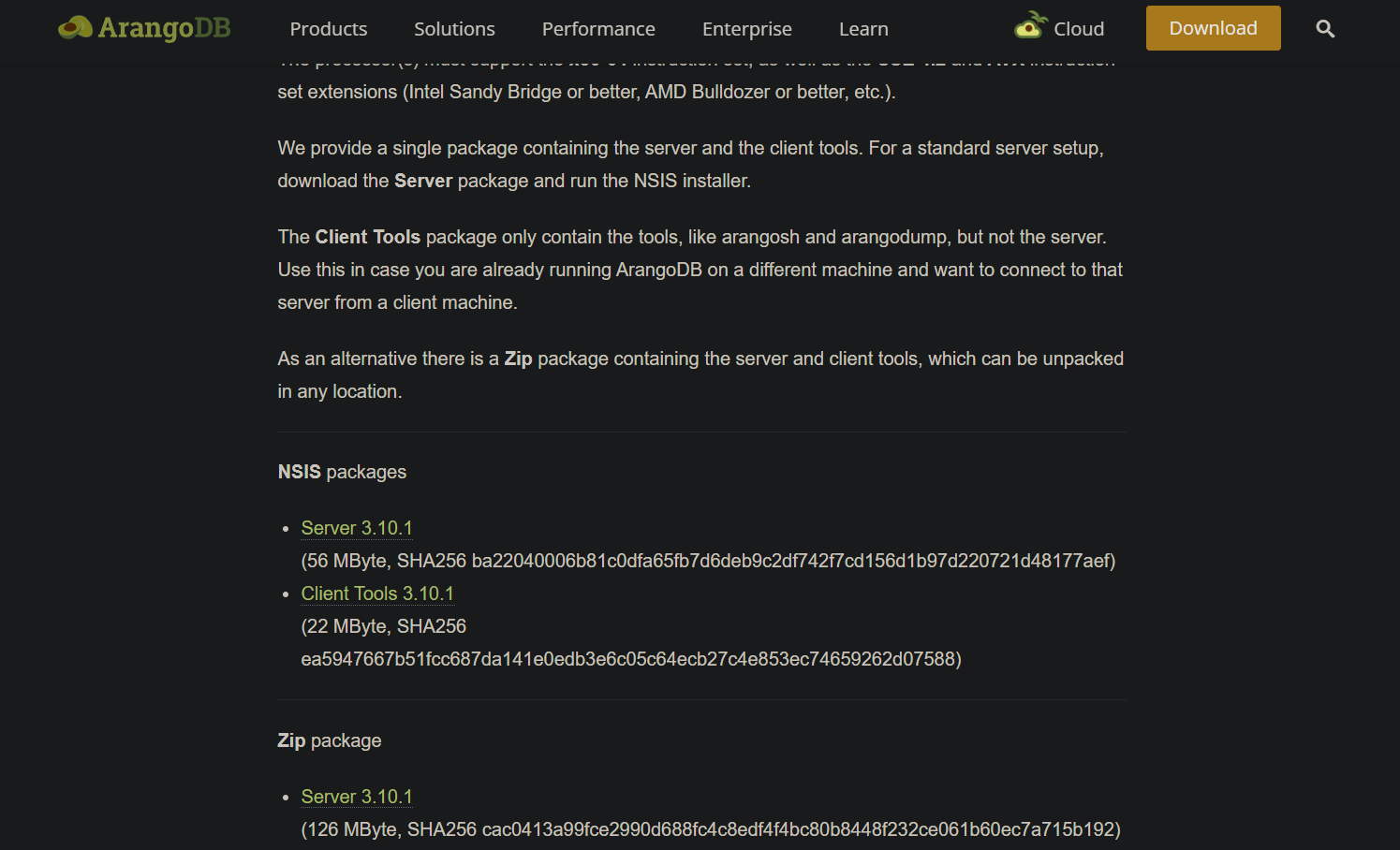
****

1. **Đặc điểm của Arangodb**

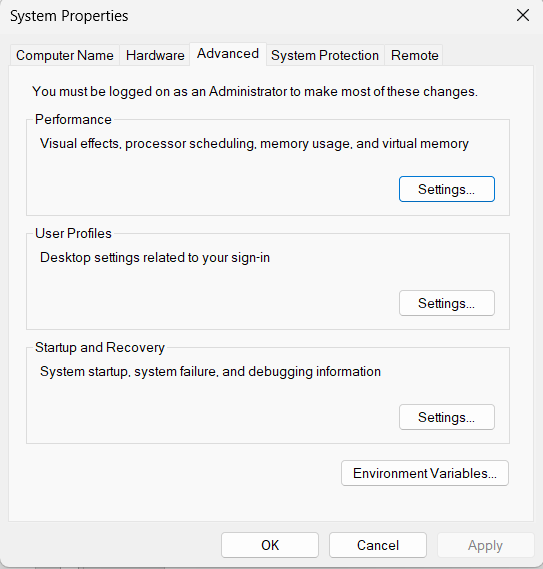
* **Nó có phiên bản Cộng đồng và phiên bản Doanh nghiệp**
* **các truy vấn có thể mở rộng khi làm việc với dữ liệu đồ họa.**
* **sử dụng JSON làm định dạng lưu trữ mặc định**
* **ban đầu bạn có thể lưu trữ một đối tượng JSON lồng nhau dưới dạng mục nhập dữ liệu trong một bộ sưu tập**
* **được chứng nhận cho hệ điều hành trung tâm dữ liệu (DC/HĐH)**
* **tích hợp với các vi dịch vụ JavaScript gốc trực tiếp trên đầu trang**
* **Nó có AQL của riêng nó (Ngôn ngữ truy vấn ArangoDB)**
* Công cụ tìm kiếm kết hợp khả năng truy xuất Boolean với các thành phần phân loại tổng quát cho phép truy xuất dữ liệu dựa trên mô hình không gian vectơ chính xác.

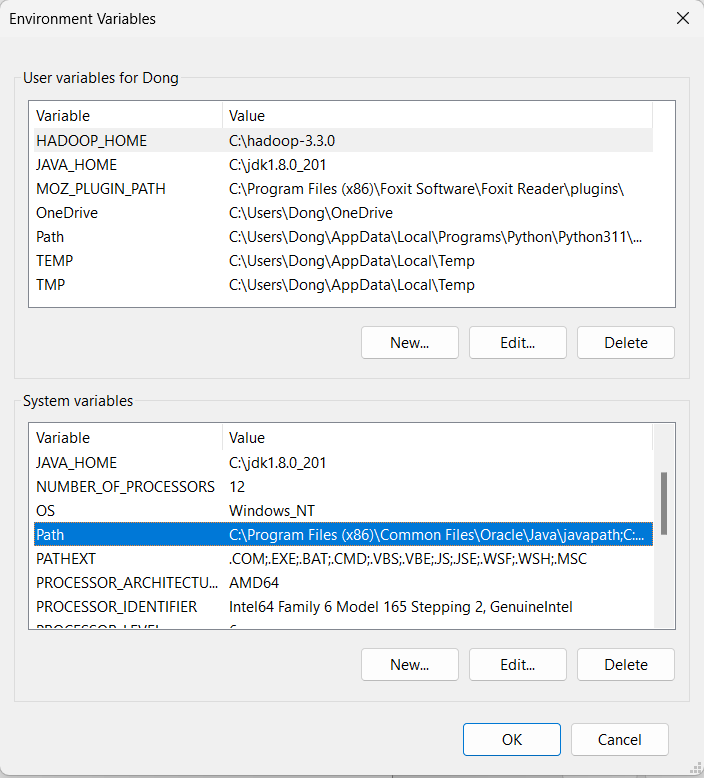
1. **Cài đặt và cấu hình cho Arangodb**

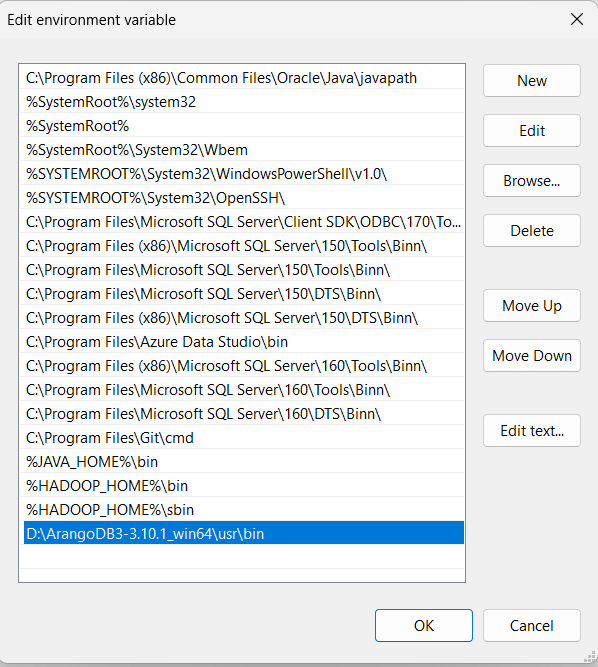
**Truy cập** [**https://www.arangodb.com/download/**](https://www.arangodb.com/download/) **để tải Arangodb**

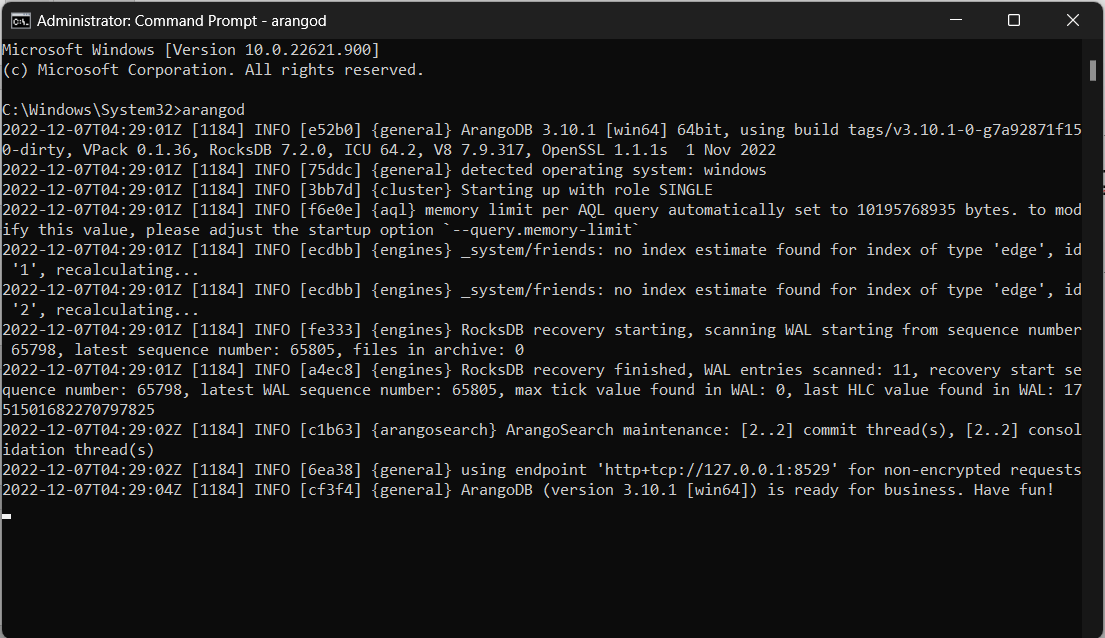
****

* **Sau khi giải nén file tải về xong, ta tiến hành setup biến môi trường cho Arangodb.**

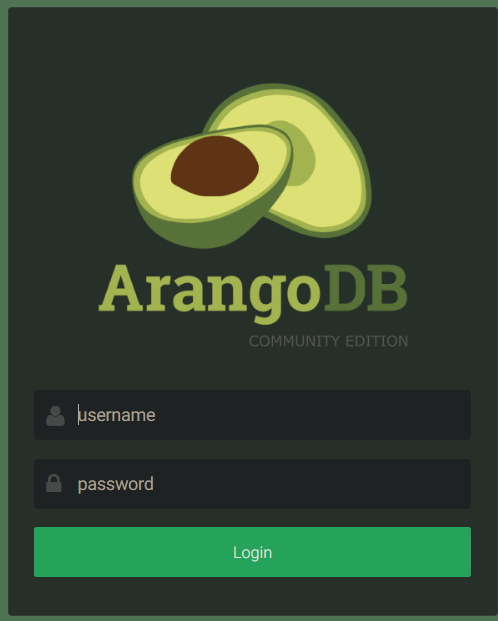
****

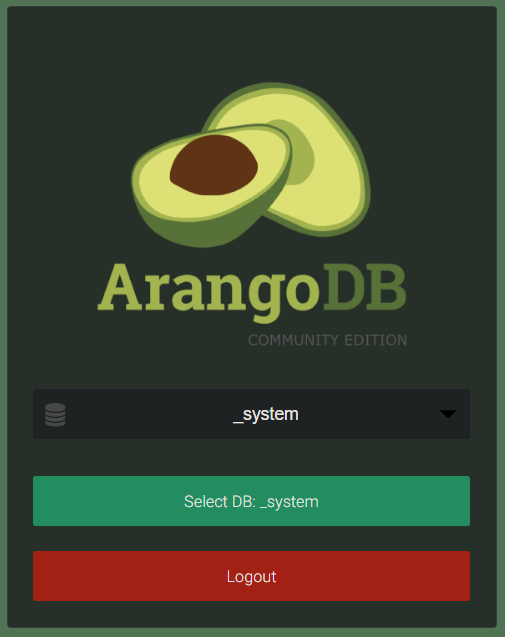
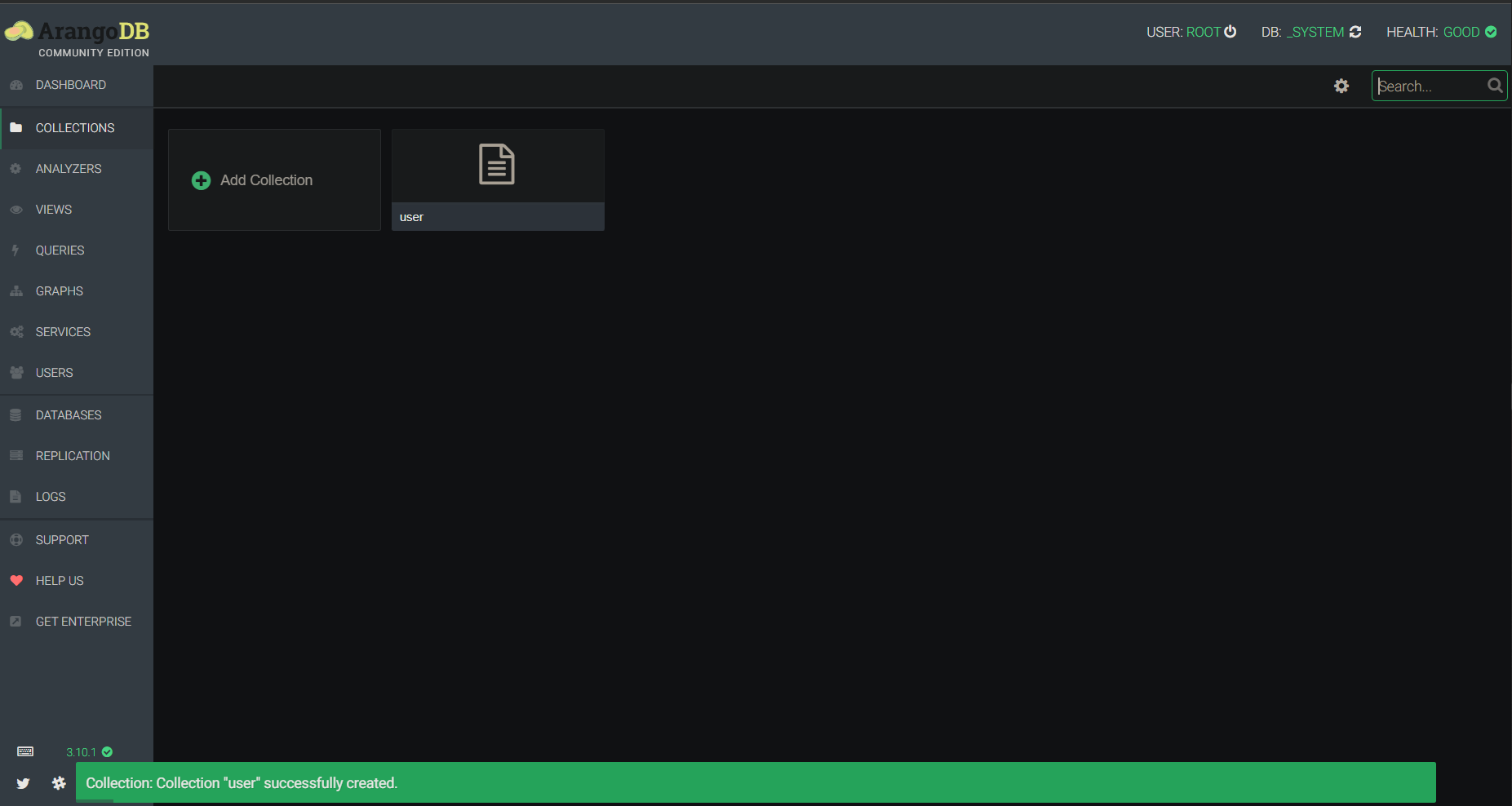
****

* **Tiếp theo ta dán đường dẫn vào path của biến môi trường, và chọn OK. Vậy là đã cấu hình xong.**
* ****
* **Ta mở CMD dưới quyền admin và nhập arangod để mở giao diện website local của Arangodb.**

****

* **Giao diện Web của Arangodb**

****

* **Truy cập localhost:8529**
* **Đăng nhập bằng tài khoản Root, không cần mật khẩu.**
* **Giao diện đăng nhập bằng tài khoản root của Arangodb**
* ****
* **Giao diện tạo thành công collection của Arangodb**
* ****
* **Phần iii: demo code**
* **1 Chèn dữ liệu vào collection**

INSERT { \_key: "9915", name: "Dong", age: 24 } INTO user

INSERT { \_key: "9916", name: "Viet", age: 20 } INTO user

INSERT { \_key: "9917", name: "Hiep", age: 20 } INTO user

INSERT { \_key: "9918", name: "Van", age: 20 } INTO user

INSERT { \_key: "9919", name: "Tuan", age: 20 } INTO user

**Trả về dữ liệu có trong collection**

RETURN DOCUMENT( ["user/9915", "user/9916", "user/9917"] )

**Trả về tất cả dữ liệu có trong collection**

FOR user1 IN user

  RETURN user1

**Sắp xếp dữ liệu thông qua các key**

FOR user1 IN user

  SORT user1.\_key

  RETURN user1

**Lọc dữ liệu theo trường dữ liệu trong collection**

  FOR user1 IN user

  SORT user1.age

  RETURN user1

FOR user1 IN user

  FILTER user1.age > 20

  SORT user1.age

  RETURN user1

**Cập nhật dữ liệu trường này cho đối tượng có key**

UPDATE "9915" WITH { age: 40 } IN user

**Nối thành chuỗi**

  FOR user1 IN user

  RETURN CONCAT(user1.name, "'s age is ", user1.age)

**So sánh các cặp dữ liệu với nhau**

  FOR user1 IN user

  FOR user2 IN user

    FILTER user1 != user2

    RETURN [user1.name, user2.name]

**Lọc ra dữ liệu các cặp với nhau**

FOR user1 IN user

  FOR user2 IN user

    FILTER user1 != user2

    RETURN {

        pair: [user1.name, user2.name],

        sumOfAges: user1.age + user2.age

    }

**Xóa dữ liệu có Key**

REMOVE "99" IN user

**Xóa dữ liệu có điều kiện**

FOR user1 IN user

    FILTER user1.age >= 30

    REMOVE user1 IN user

**Phần 3: Kết Luận**

Công trình này đã xem xét và nghiên cứu dữ liệu lớn trong thời gian gần đây đã và đang có cách xử lý ngày càng tăng đặc biệt trong khối lượng dữ liệu. Cơ sở dữ liệu có cấu trúc gặp khó khăn khi xử lý dữ liệu phi cấu trúc, do kích thước của nó. Để hiểu được dữ liệu lớn, kiến ​​trúc và phương pháp mới là cần thiết, công việc này cũng đã kiểm tra các loại Kiến trúc cơ sở dữ liệu NoQuery dữ liệu lớn khác nhau liên quan đến chúng, tầm quan trọng và cách sử dụng.